

2010

Power Envelope Expansion using a Solidity Matching Scheme for a Circulation Controlled Vertical Axis Wind Turbine

Jay Paul Wilhelm
West Virginia University

Follow this and additional works at: <https://researchrepository.wvu.edu/etd>

Recommended Citation

Wilhelm, Jay Paul, "Power Envelope Expansion using a Solidity Matching Scheme for a Circulation Controlled Vertical Axis Wind Turbine" (2010). *Graduate Theses, Dissertations, and Problem Reports*. 4672.

<https://researchrepository.wvu.edu/etd/4672>

This Dissertation is protected by copyright and/or related rights. It has been brought to you by the The Research Repository @ WVU with permission from the rights-holder(s). You are free to use this Dissertation in any way that is permitted by the copyright and related rights legislation that applies to your use. For other uses you must obtain permission from the rights-holder(s) directly, unless additional rights are indicated by a Creative Commons license in the record and/ or on the work itself. This Dissertation has been accepted for inclusion in WVU Graduate Theses, Dissertations, and Problem Reports collection by an authorized administrator of The Research Repository @ WVU. For more information, please contact researchrepository@mail.wvu.edu.

Power Envelope Expansion using a Solidity Matching Scheme for a Circulation Controlled Vertical Axis Wind Turbine

by

Jay Paul Wilhelm

Dissertation Submitted to the College of Engineering and Mineral Resources at West Virginia University in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Mechanical Engineering

James E. Smith, Ph.D., Chair

Larry E. Banta, Ph.D.

Powsiri Klinkhachorn, Ph.D.

Jacky C. Prucz, Ph.D.

Roy S. Nutter, Ph.D.

Department of Mechanical and Aerospace Engineering

Morgantown, West Virginia

2010

Constant rotational speed vertical axis wind turbines (VAWTs) with fixed blades can have a limited performance envelope as the wind speed changes. Unlike horizontal axis wind turbines that can change the blade pitch to compensate for wind speed changes, vertical axis wind turbines are currently unable to achieve a blade pitch change reliably. A different method of changing the blade's lift characteristics is needed by vertical axis wind turbines to increase its power envelope.

Circulation control (CC) can change the characteristics of an aerofoil by expelling air, moving at multiple times the free stream velocity, out of a thin slot along a rounded trailing edge. This jet of air bends around the rounded edge and entrains the free stream air, which ultimately increases lift. The use of an upper slot increases lift for positive angles of attack, while another slot on the lower surface can be used for negative angles of attack. The opposite may also be applied for a resultant decrease in lift. These slots are controllable by a valve inside the blade and can be turned on and off many times during rotation. The amount of lift increase due to circulation control is roughly proportional to the mass flow and velocity of the jet, within blade stall.

The ability to control lift, depending upon the amount of blowing, was applied to the blade of a vertical axis wind turbine to control lift generation and thus affect the power envelope. The solidity, or the ratio of the number of blades times the chord divided by the radius, greatly affects the performance of a turbine. A low solidity out-performs a high solidity rotor at low tip speed ratios and the opposite for mid-to-high tip speed ratios. Analyzing simulations of a VAWT with the same radius but different solidities, it was found that the solidity and tip speed ratio are the determining factors in performance. Circulation control can, for a VAWT, change the virtual chord and in effect both increase and decrease the virtual solidity. To mimic the solidity from a given initial solidity, circulation control was used to match the performance during rotation. The result is a VAWT that can change its apparent solidity depending upon the tip speed ratio.

An analytical model of a vertical axis wind turbine with valved circulation controlled dual slots was created and used to develop a control scheme to increase a VAWT's performance envelope. A control scheme was then developed and simulated which changed the apparent solidity of the CC-VAWT depending upon the wind speed which resulted in an increase of about 22% in overall power capture.

Acknowledgements

I would like to thank Dr. James E. Smith for taking me under his wing and believing that I could accomplish a Ph.D. I would also like to thank Dr. Roy Nutter for always having kind words of motivation and Dr. Powsiri Klinkhachorn who always lifted my spirits with a shared enthusiasm for new technological gadgetry. Finally, credit goes to Dr. Larry Banta and Dr. Jacky Prucz for supporting me and being on my committee.

I would also like to thank Dr. Franz ‘Andy’ Pertl for always being scientific in his unwavering quest for knowledge, Dr. Gerald ‘Bear’ Angle for his vast understanding of aerodynamics, and Emily Pertl, soon to be Ph.D., for holding us all together.

The other graduate students, also suffering under Dr. Smith, deserve a nod of thanks especially, Jonathan Kweder, Chad Panther, and Kenny Williams.

Members of my family, my mother Carol and brother Dr. Thomas E. Wilhelm, also deserve a special thanks for always listening and providing me with support. Lastly, I would like to thank my wonderful wife, Charity, for her overwhelming support, patience with me, and wonderful home cooked meals.

I would like to dedicate this document to my late father, Paul J. Wilhelm (1942-2001), who left us early but made a large impression on me and made me the person I am today.

Table of Contents

Abstract	ii
Acknowledgements	iii
Table of Figures	vii
Chapter 1. Introduction	1
1.1 Research Objective	6
Chapter 2. Literature Review	7
2.1 Vertical Axis Wind Turbines	7
2.1.1 VAWT Blade	7
2.1.2 Flow Curvature	8
2.1.3 General Aerodynamic Forces Generated by a VAWT	9
2.1.4 Power Output for a VAWT	14
2.1.5 VAWT Basics Summary	15
2.2 VAWT Modeling	15
2.2.1 Momentum Models	16
2.2.2 Vortex Model	21
2.2.3 Dynamic Stall Modeling	23
2.2.4 Wind Distribution Modeling	27
2.2.5 VAWT Aerodynamic Modeling Summary	30
2.3 Circulation Control	30
2.4 Wind Turbine Blades	32
2.4.1 Fixed Blade Pitch Effect on Performance	33
2.4.2 VAWT Blade Summary	36
2.5 Controlling Wind Turbines	36
2.5.1 Methodology for Controller Development	37
2.5.2 VAWT Control Techniques	37
2.5.3 Speed and Torque Control	39
2.5.4 Controlling the CC-VAWT	42
2.5.5 Controlling Wind Turbines Summary	42
2.6 Literature Review Summary	43
Chapter 3. CC-VAWT Analytical Modeling	44
3.1 Introduction	44
3.2 The Double Multiple Stream Tube Model	44
3.2.1 Upstream Solution	45
3.2.2 Downstream Solution	49
3.2.3 Finding the Induction Factor	51
3.2.4 Coefficient of Performance	51
3.2.5 Circulation Control applied to a VAWT in the Momentum Model	52
3.2.6 Momentum Model Simulation Results	54
3.2.7 Momentum Model Validation	57
3.2.8 Breakdown Issues	58
3.2.9 Momentum Model Summary	59
3.3 Two Dimensional Vortex Modeling	60
3.3.1 Model Implementation	60
3.3.2 Modeling of Airfoil Lift and Drag as complex values	61
3.3.3 Circulation and lift	62
3.3.4 Complex Velocity Induced from a vortex	63
3.3.5 Model Convergence	67
3.3.6 Dynamic Stall in the Vortex Model	67

3.3.7 Vortex Model Validation	68
3.3.8 Circulation Control applied to a VAWT in the Vortex Model	70
3.3.9 Vortex Model Simulation Results	72
3.3.10 Impacts of changing the Step Size $\Delta\theta$	77
3.3.11 Steady State Convergence	78
3.3.12 2D Vortex Analytical Model Summary	79
3.4 Three Dimensional Vortex Modeling.....	80
3.4.1 3D Vortex Model Summary.....	86
3.5 Vortex Model Execution Time Decrease	86
3.5.1 CUDA	87
3.5.2 Computational Speed Increase Results.....	88
3.5.3 Vortex Model Computation Time Reduction Summary.....	90
3.6 CC-VAWT Modeling Conclusion	91
Chapter 4. CC-VAWT Blade Characteristics.....	92
4.1 Introduction.....	92
4.2 Measurement of Lift and Drag	92
4.3 CC-NACA0018 Steady State Model	94
4.4 The CC-NACA0018 Rapid Prototype Model.....	95
4.5 Lift and Drag Coefficients for the CC-NACA0018	97
4.6 Lift Time Response of the CC-NACA0018	99
4.7 Comparison of the CC-NACA0018 to CC-Ellipse	101
4.7.1 CC-VAWT Blade	104
4.8 The Jet	105
4.9 CC-VAWT Blade Characteristics.....	106
4.10 Experimental CC-VAWT Development.....	131
Based upon the preliminary and wind tunnel airfoil data,	131
4.11 Data Acquisition.....	134
4.12 Tangential Force Predictions.....	135
4.13 Experimental Apparatus Conclusion.....	136
Chapter 5. Solidity Matching Scheme Using Circulation Control for a VAWT	107
5.1 Introduction.....	107
5.2 Constant and Variable Rotation Speed	107
5.3 Rotational Blowing Scheduling	108
5.3.1 Summary of Steps Required to Match Solidity for Maximum C_p	121
5.4 Evaluation of the Solidity Matching Scheme for Achieving a Maximum C_p	122
5.5 Energy Capture as Determined by a Wind Distribution	124
5.6 Power Calculation of the CCVAWT	125
5.7 Evaluation and Performance for a Wind Distribution of the Solidity Matching Scheme	126
5.8 Blowing Coefficient Required to Match Solidity	128
5.9 Circulation Control Utilization Summary	130
Chapter 6. Conclusion	137
6.1 Research Conclusion	138
6.2 Recommendations	138
References.....	140
Appendix A. Source Code Listing.....	144
A.1 VortexModelComplex.....	144
A.2 VortexModel3D	148
A.3 CCVAWTController.....	156
A.4 MomentumModel	164

A.5 VelFillament	169
A.6 VelBLayer	170
A.7 ModelDataHolder	170
FindCuCtNeededToMatch	171
A.8 FindPowerForCCEllipse	172
A.9 FindSolAndWRForCCEllipse	174

Table of Figures

Figure 1-1 Horizontal Axis Wind Turbine (Monster 2008)	2
Figure 1-2 Vertical Axis Wind Turbine (Wolfe 1981).....	3
Figure 2-1 Free-body diagram for an airfoil (Anderson 2007).....	8
Figure 2-2 Flow Curvature	9
Figure 2-3 NACA0018 Cross Section	10
Figure 2-4 Coefficient of Lift and Drag for NACA0018	11
Figure 2-5 Angle-of-Attack as determined by TSR and ϑ (Pawsey 2002)	12
Figure 2-6 V_{rel} as function of rotation position and TSR (Kirke 1998)	13
Figure 2-7 Forces on an example VAWT vs Azimuth and TSR (Pawsey 2002)	14
Figure 2-8 The Stream tube model (Islam 2006)	16
Figure 2-9 Multiple Stream Tube Model (Islam 2006)	19
Figure 2-10 The Double Stream Tube Model (Pawsey 2002).....	20
Figure 2-11 Vortex Model Shedding (Strickland 1980)	22
Figure 2-12 Dynamic Stall (Pawsey 2002)	24
Figure 2-13 Two Bladed VAWT Blade Force and Measurements, TSR of 3 (Pawsey 2002)	27
Figure 2-14 Rayleigh Distribution and Power of the Wind	29
Figure 2-15 Lift & Drag Coefficients of a CC 23% Ellipse within Stall.....	32
Figure 2-16 Predicted Performance of Variable Pitch VAWT (Pawsey 2002)	33
Figure 2-17 Time Response of Lift (Angle 2008)	34
Figure 2-18 Lift Coefficient of Blade with Blowing Slot (Angle 2008)	35
Figure 2-19 Possible Blade Blowing Slot Configuration	36
Figure 2-20 Ideal HAWT Power Curve	37
Figure 2-21 Speed Controller (Bianchi 2007)	40
Figure 2-22 Torque Controller (Bianchi 2007)	40
Figure 2-23 Hybrid Speed/Torque Controller (Bianchi 2007).....	40
Figure 2-24 Pitch Control Mod I (Staelens 2003)	42
Figure 2-25 Pitch Control Mod II (Staelens 2003)	42
Figure 2-26 Pitch Control Mod III (Staelens 2003)	42
Figure 2-27 Pitch Control Power (Staelens 2003)	42
Figure 3-1 Relative blade velocities and angles of attack at various tip speed ratios λ and blade positions ϑ	45
Figure 3-2 Multiple Stream Tube Velocity Boundaries	46
Figure 3-3 Double Multiple Stream Tube Velocity Boundaries	50
Figure 3-4 Lift Coefficient of Estimated CC-VAWT blade and NACA0018.....	53
Figure 3-5 Drag Coefficient of Estimated CC-VAWT blade and NACA0018	53
Figure 3-6 CC-VAWT $C_{\mu} = 1\%$ Relative Velocity During Rotation.....	54
Figure 3-7 CC-VAWT $C_{\mu} = 1\%$ Angle of Attack.....	55
Figure 3-8 CC-VAWT $C_{\mu} = 1\%$ Induction Factors.....	55
Figure 3-9 $C_{\mu} = 1\%$ Torque Force	56
Figure 3-10 $C_{\mu} = 1\%$ Coefficient of Performance During Rotation.....	56
Figure 3-11 $C_{\mu} = 1\%$ Coefficient of Performance	57
Figure 3-12 Comparison of developed model to literature model (Strickland 1975).....	58
Figure 3-13 Performance Curves for NACA0018 Momentum Model.....	59
Figure 3-14 Non-dimensional VAWT Geometric Setup in the Complex Plane	61
Figure 3-15 Vortex Simulation of a Turbine Showing Free Convected Vortices and Blade Bound Vortices.....	63
Figure 3-16 Induced Velocity from a Viscous Core Vortex at a Distance Δp	65
Figure 3-17 Vortex Model Compared to VDART2 (Strickland 1980)	69
Figure 3-18 Comparison of NACA0018 Performance for Momentum, Vortex, and VDART2 Models	70
Figure 3-19 Lift Coefficient of Estimated CC-VAWT blade and NACA0018	71
Figure 3-20 Drag Coefficient of Estimated CC-VAWT blade and NACA0018	71
Figure 3-21 Vortex Model Relative Velocity TSR=4.....	72
Figure 3-22 Vortex Model Angle of Attack TSR=4.....	72

Figure 3-23 Vortex Model Tangential Force TSR=4	73
Figure 3-24 Coefficient of Performance.....	73
Figure 3-25 Vortex Model CC-VAWT $C_{\mu}=0\%$ Performance.....	74
Figure 3-26 Vortex Model CC-VAWT $C_{\mu}=1\%$ Performance.....	74
Figure 3-27 Vortex Model CC-VAWT $C_{\mu}=10\%$ Performance.....	75
Figure 3-28 Vortex Model Velocity Field.....	76
Figure 3-29 Vortex Model Velocity Field with Blade Positions	76
Figure 3-30 Coefficient of Performance for Select $\Delta\vartheta$	77
Figure 3-31 Normalized Execution Time for Varying $\Delta\vartheta$	78
Figure 3-32 Field of Turbulent VAWT	79
Figure 3-33 Cp Convergence of the Vortex Model.....	79
Figure 3-34 Geometry to calculate induced velocity from a vortex line element	82
Figure 3-35 Non-dimensional 3D VAWT geometric setup	83
Figure 3-36 Blade subdivision into blade elements and associated vortex line elements.	84
Figure 3-37 Constant velocity magnitude surfaces at 0.95, 0.9 and 0.85 V_{∞} for $\lambda=4$	85
Figure 3-38 Vortex filament wake for TSR of 4 after 5 revolutions	85
Figure 3-39 VAWT Coefficient of Performance Comparison of 2D & 3D Vortex Model.....	86
Figure 3-40 Flow Chart of CUDA Vortex Model Speedup.....	88
Figure 3-41 Required Computing Time for the 2-D Vortex Model	89
Figure 3-42 Required Computing Time for the 3-D Vortex Model	90
Figure 4-1 Model Stand with Model and Wind Tunnel Test Section.....	93
Figure 4-2 Wind Tunnel Model Support and Mounting System	93
Figure 4-3 Model Support Setup with Bearing	94
Figure 4-4 CC-NACA0018 CAD Drawing of the Metal Support Structure.....	95
Figure 4-5 CC-NACA0018 Assembled Support Structure.....	95
Figure 4-6 CC NACA 0018 Model Mounted in Wind Tunnel.....	95
Figure 4-7 CC-NACA0018 Cross Section	96
Figure 4-8 CAD Drawing of the CC-NACA0018 Rapid Prototype Machine.....	96
Figure 4-9 CC-NACA0018 Rapid Prototype Blade Skin & Internal Components.....	97
Figure 4-10 Coefficient of Lift for the CC-NACA0018	98
Figure 4-11 Coefficient of Drag for the CC-NACA0018	98
Figure 4-12 Tangential Coefficient of the CC-NACA0018.....	99
Figure 4-13 Lift & Drag Response at AoA of 0 deg, C_{mu} 5% and Reynolds number of 300k.....	100
Figure 4-14 Lift & Drag Response at AoA of 0 deg, C_{mu} 10% and Reynolds number of 300k.....	101
Figure 4-15 Cross Section of a 23% Elliptical Airfoil.....	102
Figure 4-16 NACA0018 Cross Section	102
Figure 4-17 Lift Coefficient of the CC-NACA0018 & CC-Ellipse.....	103
Figure 4-18 Drag Coefficient of the CC-NACA0018 & CC-Ellipse.....	103
Figure 4-19 Tangential Coefficient of the CC-NACA0018 & CC-Ellipse	104
Figure 4-20 Cross Section of the CC-VAWT Elliptical Blade.....	105
Figure 4-21 Circulation Control Turbine Blade (Transparent Skin)	132
Figure 4-22 CAD Drawing of the Experimental Apparatus (Dimensions in meters)	133
Figure 4-23 Experimental CC-VAWT Apparatus	133
Figure 4-24 Experimental Tangential Force at TSR of 4.....	136
Figure 5-1 Simulated Performance with Constant Blowing Coefficient at solidity of 0.1	110
Figure 5-2 Ideal HAWT Power Curve	111
Figure 5-3 Performance of VAWT for Multiple Solidities.....	112
Figure 5-4 CC-VAWT Blade, 23% Ellipse	115
Figure 5-5 Lift & Drag Coefficients of a 23% Elliptical Aerofoil with Circulation Control within stall	116
Figure 5-6 Lift & Drag Coefficients of a 23% Elliptical Aerofoil with Circulation Control.....	116
Figure 5-7 Tangential Coefficient of the CCVAWT Blade	117
Figure 5-8 Augmentation Ratio for Increasing Blowing of a 20% Ellipse (Abramson 1974).....	118
Figure 5-9 Required C_t for Solidity Matching	119

Figure 5-10 a) c_t During Rotation to Match a Solidity of 0.2 from 0.1 and b) c_{μ} required to Achieve Performance at Solidity of 0.2 from 0.1	120
Figure 5-11 Coefficient of Performance using Solidity Matching	121
Figure 5-12 Simulated Coefficient of Performance of the Solidity Matching Scheme for $\sigma_0 = 0.1$	123
Figure 5-13 Simulated Coefficient of Performance of the Solidity Matching Scheme for $\sigma_0 = 0.15$	123
Figure 5-14 Rayleigh Distribution and Power of the Wind	125
Figure 5-15 a) Simulated Power Capture for VAWT with 23% Ellipse Blades and b) C_p curve at solidity of 0.17	127
Figure 5-16 Simulated Power Captured for a CC-VAWT at 7 m/s Average Wind	128
Figure 5-18 Blowing Coefficient Required from an Initial Solidity of 0.15 to Match Best Solidity	129
Figure 5-19 Average Blowing Coefficient Required for Initial Solidities	130

Chapter 1. Introduction

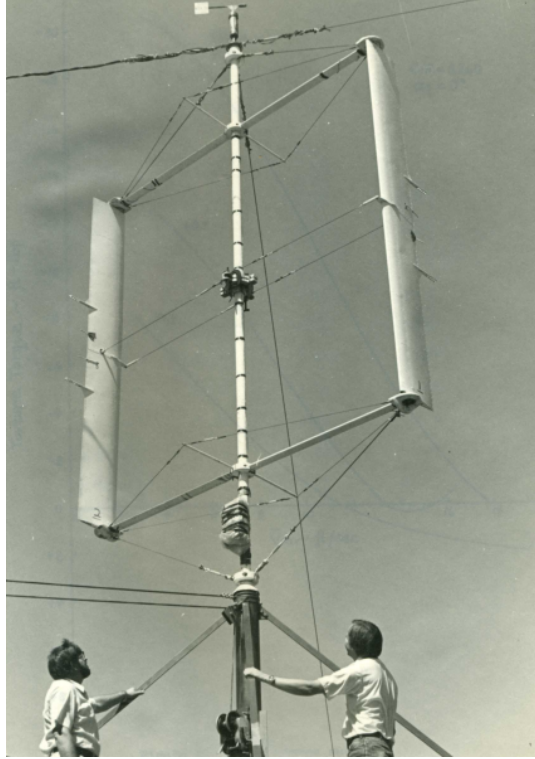
Wind turbines can provide an inexpensive, localized, renewable energy source and while intermittent with the wind, they can supplement an existing supply grid or provide a standalone source of power. Unlike typical power producers that rely on nonrenewable resources, wind turbines do not pollute nor do they consume resources during power production. Additionally their installation and maintenance costs are small compared to the more traditional power producers. Developing areas and power expansion projects could benefit from the simplicity of wind turbine installation versus the construction of the infrastructure for a nonrenewable resource power plant.

Generation of power from wind can occur due to a wing-like device that captures and converts the air's moving kinetic energy into rotational movement, which is then converted to mechanical work or to generate electrical energy. This process has been occurring for thousands of years in many different forms through wind turbines. The earliest wind turbines turned a large mill wheel for the crushing of grains. Recently, wind turbines have become much more advanced and more attention has been focused towards engineering them with higher overall efficiencies.

While the benefits of wind turbines are substantial, some shortcomings do exist. First, when the wind is not blowing fast enough to meet design requirements, little or no power can be produced. Second, there is a limit to the amount of energy that can be extracted from wind which usually occurs over a small range of wind speeds. Even with these limitations, wind turbines are still highly sought after as a renewable energy source.

Currently, wind turbines are divided into two types. The most popular is the HAWT (Horizontal Axis Wind Turbine), shown in Figure 1-1. This type is similar to a propeller, except that it is driven by the





produce more lift without physically changing the airfoil or its angle-of-attack (Angle 2008). This technology when applied to a VAWT could reduce or eliminate the need for blade pitch actuation that may be required for increased energy capture. Improvements to the power generated by increasing the lift of the blade using circulation controlled blades over a larger wind speed range would then be possible.

Wind turbine performance is measured by the coefficient of performance (C_p) and the overall power captured. The coefficient of performance for a wind turbine has an upper theoretical limit as given by the Betz Law. Fixed angle-of-attack VAWTs will typically only have high performance around a single wind speed as with HAWTs. Variable angle-of-attack VAWTs can increase the maximum performance range over that realized by a fixed VAWT and can also extend the performance for broader wind speeds. A CC-VAWT will be able to improve overall energy capture over a fixed bladed VAWT without the use of a complex mechanical pitch control system as will be shown in this study. Such a novel method of using circulation control on a VAWT will be unlike any other system that has been created to date because of its ability to increase or decrease the amount of lift without physically changing the blade. This will all be possible with the help of a control scheme which manages the blowing slots during rotation for maximum power capture at a range of wind speeds.

Development of an energy capture scheme for blowing slot control of a CC-VAWT will explore the following topics. First, examination of the current state of the art wind turbine modeling and aerodynamic control systems will be reviewed. Second, aerodynamic modeling techniques will be documented and their implementation results will be compared to literature models and experimental data. Third, a CC-VAWT blade will be looked at in depth and evaluated for its lift, drag, and tangential force capabilities and the augmentation factor of circulation control. Finally, a blowing slot control scheme will be developed and performance predictions will be made.

State of the art wind turbine topics present in literature such as design methods, modeling techniques, control over wind turbines and power evaluation methods will be reviewed. The wealth of knowledge present in the literature will provide a starting point for calculating how much and why energy is captured by a VAWT, how to calculate aerodynamic forces, circulation control methods used, and any other previous work associated with a VAWT.

Modeling the aerodynamics of a VAWT can involve using techniques such as a momentum analysis or the potential flow method. The momentum model is considered a basic aerodynamic model, but will be used for an initial analysis of a VAWT. The potential flow model, or vortex model, is considered more complex when compared to the momentum model and is more accurate. The potential flow model can also support advanced concepts, such as unsteady flow and blade-wake interaction. These models can predict the basic blade forces of lift and drag, the turbine torque, and the coefficient of performance can all be predicted by these models. One or both of the developed models will then be used for a control scheme design and evaluation once the model has been verified.

The characteristics of a CC-VAWT blade will determine, in combination with parameters such as solidity and wind speed, the performance or energy capture efficiency. An initial blade shape, the NACA0018 which is commonly used in VAWT, will be examined with circulation control applied to it. The lift, drag, and tangential blade forces will be compared to commonly used circulation control airfoils. From this analysis a blade shape, with the highest performance and impact with circulation control and impact of VAWT performance, will be chosen for further simulations for controller design and for use in an experimental apparatus.

Finally, once an aerodynamic analytical model has been created, a blade has been selected, and an experiment performed for model verification, a control scheme can be created. This control scheme may utilize a constant blowing coefficient or rate, or may use a variable blowing rate. The solidity,

turbine radius, and rotational speed will be determined by simulating the CC-VAWT without blowing for the maximum energy capture. Circulation control will only be used when operating outside of the design speed. The amount of blowing, or the variable amount of blowing, will be determined by the control scheme, which will be determined by simulations.

1.1 Research Objective

The objective of this research was to determine how to utilize circulation control on a vertical axis wind turbine to increase energy capture over that of a traditional (fixed bladed) VAWT. A literature review was performed, an aerodynamic analytical model of a CC-VAWT was created by building on existing VAWT analytical models to incorporate the effects of circulation control. A control scheme was then designed and simulated, based upon an analytical model, which utilized blowing to maximize energy capture of a CC-VAWT.

Equation Chapter 2 Section 1

Chapter 2. Literature Review

Vertical axis wind turbines, analytical modeling techniques, basic aerodynamics of turbine blades, and circulation control are reviewed. Control schemes for turbines and more specifically VAWT are also overviewed. The techniques and material presented here is background information which applies to the development of a control scheme for a wind turbine with circulation controlled blades with the focus on improving the capture rate of energy from the wind. [Equation Chapter 2 Section 1](#)

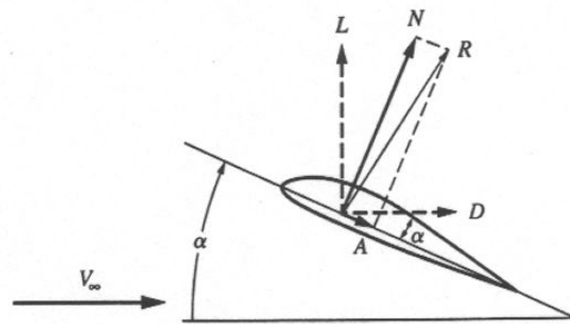
2.1 Vertical Axis Wind Turbines

VAWTs have been experimented with since the 1970's and significant findings were identified by experimental research. A VAWT blade can be susceptible to drastic changes in airfoil performance when flow curvature is present. The torque produced is not constant during rotation and must be considered during support structure design. A mechanism for changing the lift on the blades is necessary to control the speed and/or torque of the turbine. Symmetrical blades are preferred to capture power upwind and downwind of the turbine. These issues were identified but not always solved in the early technical literature and since the 1980's research on the VAWT dropped off significantly. The starting point for all VAWT design work has been the characteristics of the blade.

2.1.1 VAWT Blade

A VAWT blade is a long airfoil that orbits the center of the turbine and is either straight, curved, or a combination of straight and curved. Some VAWT have fixed angle-of-attack blades while others vary the angle depending upon wind speed, rotation speed, and rotation position. The VAWT blade is the sole force producer responsible for power generation of the turbine.

Static aerodynamic forces generated by an airfoil are a function of the air density, free-stream velocity, wind area, and aerodynamic coefficients, which are determined by the angle-of-attack. The



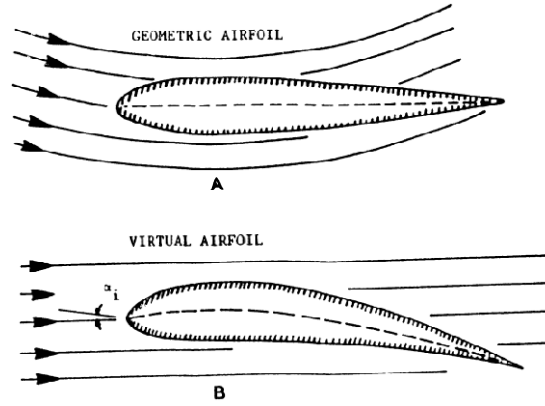


Figure 2-2 Flow Curvature

2.1.3 General Aerodynamic Forces Generated by a VAWT

The static aerodynamic forces of lift and drag generated by an airfoil can be calculated by equations (2.1) and (2.2), where ρ is the air density constant, V_{rel} is the air speed across the airfoil, and the coefficient term c_L or c_D , which is determined by the angle-of-attack and the geometric characteristics of the airfoil. The lift and drag forces are normal to each other with the drag force parallel to the relative airflow. (Anderson 2007)

$$F_L = \frac{1}{2} \rho A V_{rel}^2 c_L \quad (2.1)$$

$$F_D = \frac{1}{2} \rho A V_{rel}^2 c_D \quad (2.2)$$

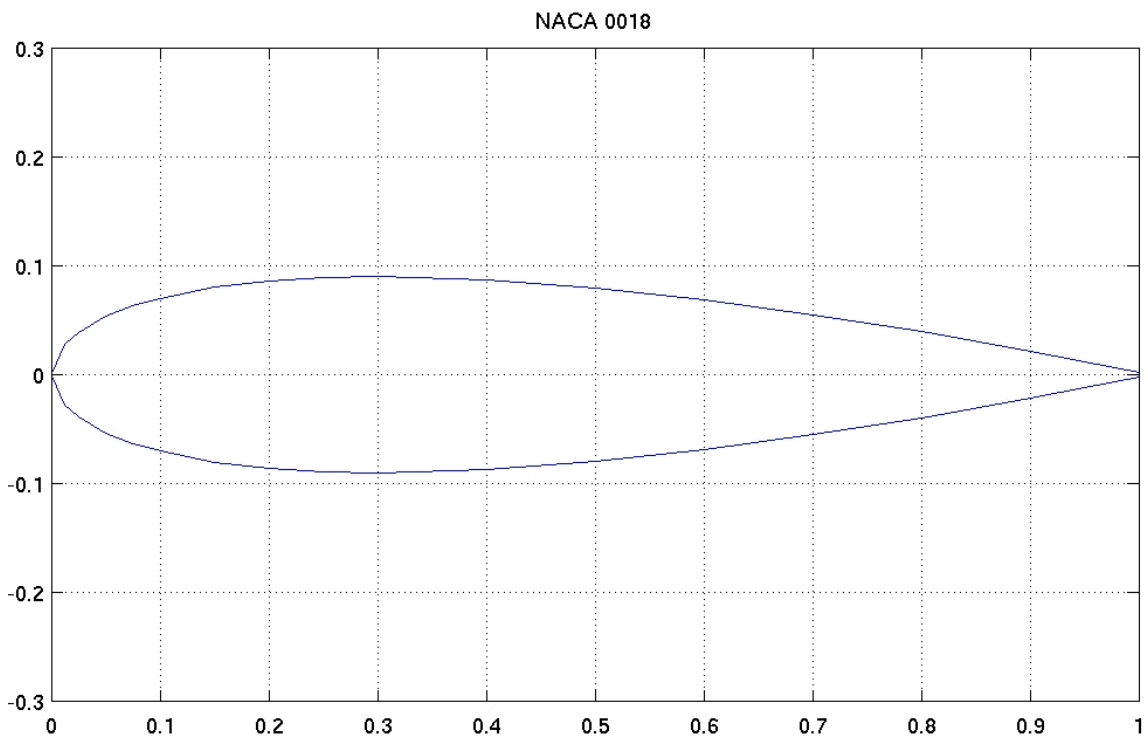
For a VAWT, the tangential force, which drives rotation, can be obtained from drag and lift which is represented by a coefficient, c_t equation(2.3). The normal force coefficient is expressed by equation(2.4). The tangential force can be calculated from (2.5) and the normal force by (2.6) where ρ is the air density, A is the blade area and V_{rel} is the velocity of the blade relative to the surrounding air. (Kirke 1998, Pawsey 2002)

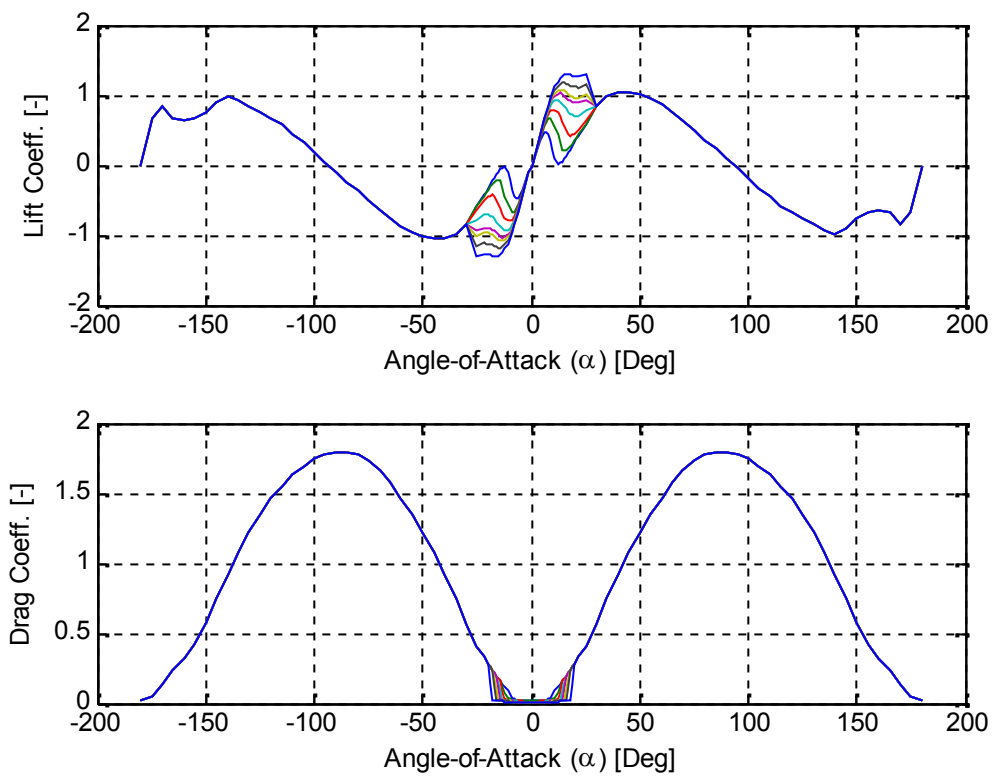
$$c_t = c_L \sin \alpha - c_D \cos \alpha \quad (2.3)$$

$$c_n = c_L \cos \alpha + c_D \sin \alpha \quad (2.4)$$

$$F_t = \frac{1}{2} \rho A V_{rel}^2 c_t$$

$$F_n = \frac{1}{2} \rho A V_{rel}^2 c_n$$

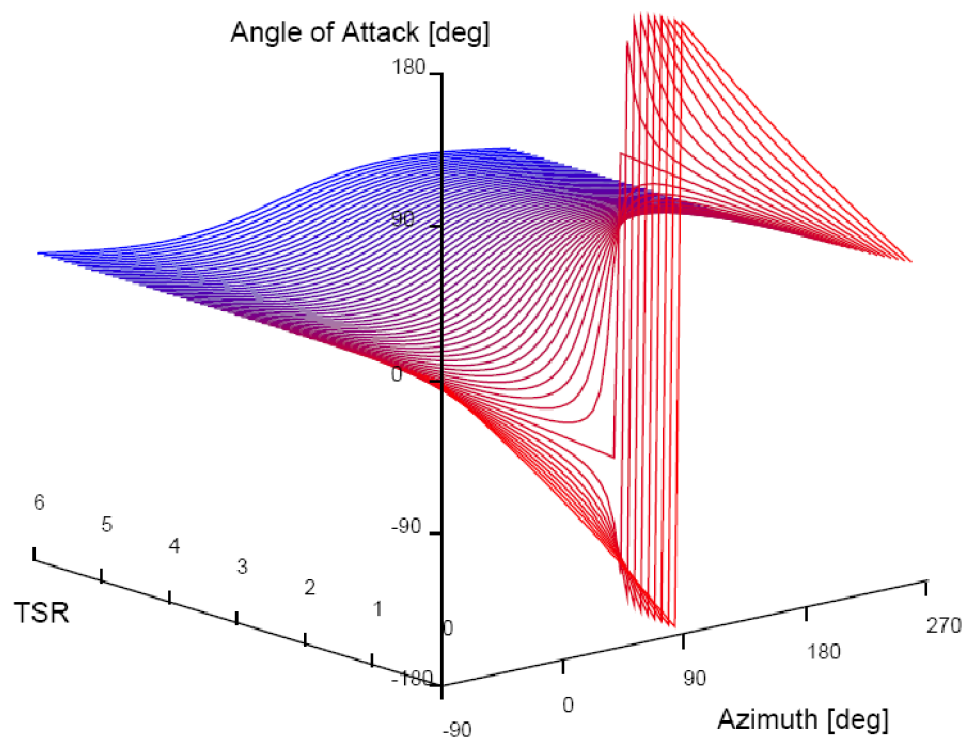




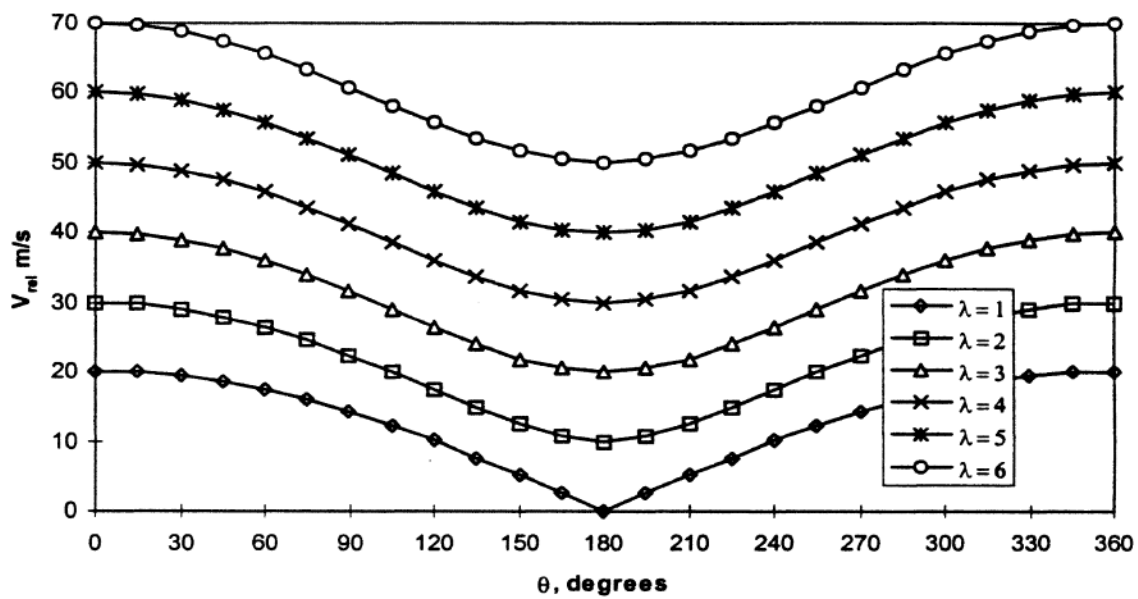
$$\lambda = \frac{\omega r}{V_\infty} = \frac{V_b}{V_\infty}$$

$$\alpha = \tan^{-1} \left(\frac{\cos \theta}{\sin \theta - \frac{\lambda}{1-a}} \right)$$

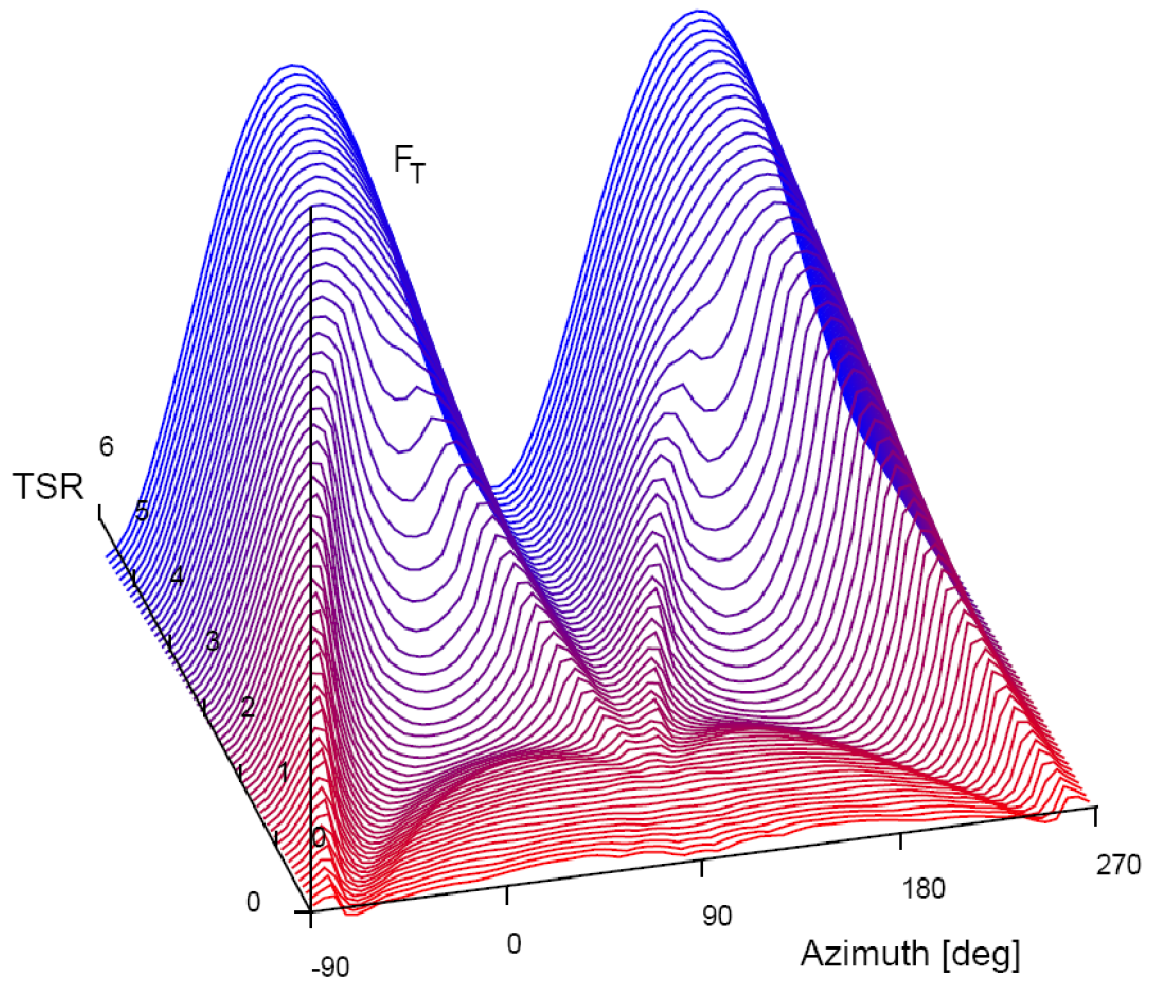
$$\sigma = \frac{N_b C}{R}$$



$$V_{rel} = V_{\infty} \sqrt{\sin^2 \theta + (\lambda - \cos \theta)^2}$$



$$\tau = \frac{F_t}{RV_\infty^2} = \frac{1}{2} \frac{C}{R} F_t$$



$$C_p = \frac{\left(1 + \frac{V_{downstream}}{V_{upstream}}\right) \left[1 - \left(\frac{V_{downstream}}{V_{upstream}}\right)^2\right]}{2} \quad (2.12)$$

Calculation of C_p from the power output of a turbine and wind velocity can be represented as

$$C_p = \frac{P_{turbine}}{0.5 \rho_{\infty} A V_{\infty}^3} \quad (2.13)$$

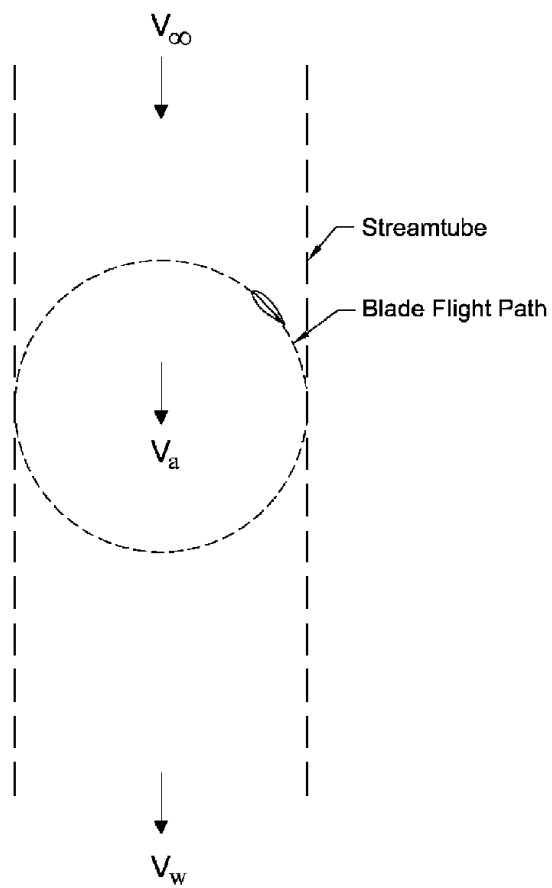
The amount of power that a wind turbine generates is the average product of torque generated (τ) and the rotational speed (ω), but is limited to a maximum efficiency commonly known as the Betz Limit. This limits the coefficient of performance to 16/27 (~0.59). This maximum occurs when the downstream velocity is 1/3 of the upstream velocity. Pitching the blades or the addition of circulation control to a VAWT will not alter the Betz limit. Lift augmentation techniques that alter the capture area would raise the power extracted while still staying within the Betz limit as it is defined.

2.1.5 VAWT Basics Summary

The technical literature provides the basic forces that are developed by the aerodynamics of a VAWT. The relationships of the forces could be used to develop a basic analytical model if an assumption is made that the inflow velocity to the VAWT is constant throughout the entire portion of rotation. To accurately model these forces, and predict the power generated, requires a more advanced analytical model, such as the momentum or vortex model.

2.2 VAWT Modeling

Analytical modeling of a system can be used as a design tool for optimal parameter selection and controller design. VAWT modeling consists of aerodynamics and wind modeling. Aerodynamic modeling can be accomplished with momentum or vortex methods. Finally, a model of the wind is necessary to accurately predict real world wind distributions.



To solve the change in momentum for a VAWT and determine the wake velocity, the induction factor must be calculated. The induction factor of a wind turbine, a , is the ratio of the reduction of the velocity in the turbine to the air velocity far away from the wind turbine and is defined as

$$a = \frac{V_{\infty} - V_1}{V_{\infty}} \quad (2.14)$$

The Momentum equation is based upon equation (2.14) and is defined as

$$V_1 = V_{\infty}(1 - 2a) \quad (2.15)$$

This equation is used to determine the wake velocity, V_1 , in the disc, where V_{∞} is the free stream velocity and a is the induction factor. The solution for the wake velocity follows an iterative procedure. This method uses the momentum equation (2.16) and a aerodynamic blade element equation (2.17), where N_B is the number of blades, C is the chord length, and V_{rel} is the relative velocity as defined in equation (2.18). The coefficients of the normal force and tangential forces are found in equations (2.4) and (2.3). An iterative procedure is used to solve for the induction factor using equation (2.19), which is common to both the momentum force equation and the aerodynamic equation. (Strickland 1975, Sharpe 1990, Spera 1994, Pawsey 2002, Leishman 2006, Dreier 2007)

$$f_{mom}^* = \pi \cos(\theta) \frac{26a + 4}{15} \quad (2.16)$$

$$f_{aero}^* = \frac{N_B C}{4\pi r} \left(\frac{V_{rel}}{V_{\infty}} \right)^2 \left(c_n - \frac{c_t}{\tan \theta} \right) \quad (2.17)$$

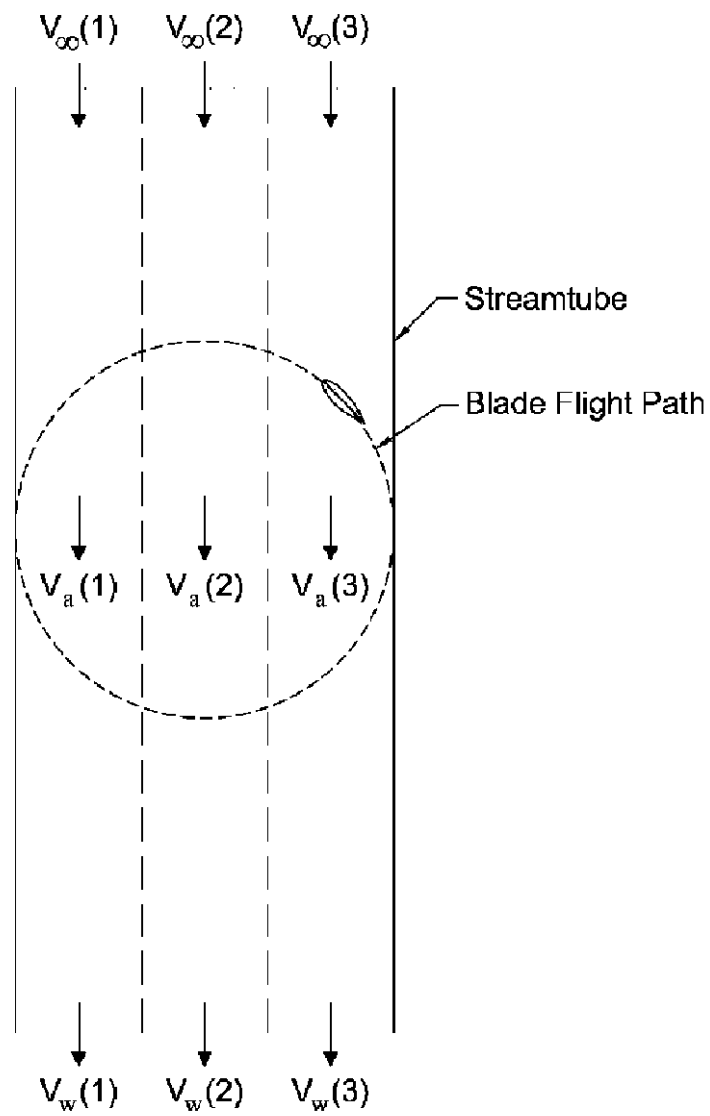
$$V_{rel} = \frac{V_{\infty} \sin \theta}{\sin \alpha} \quad (2.18)$$

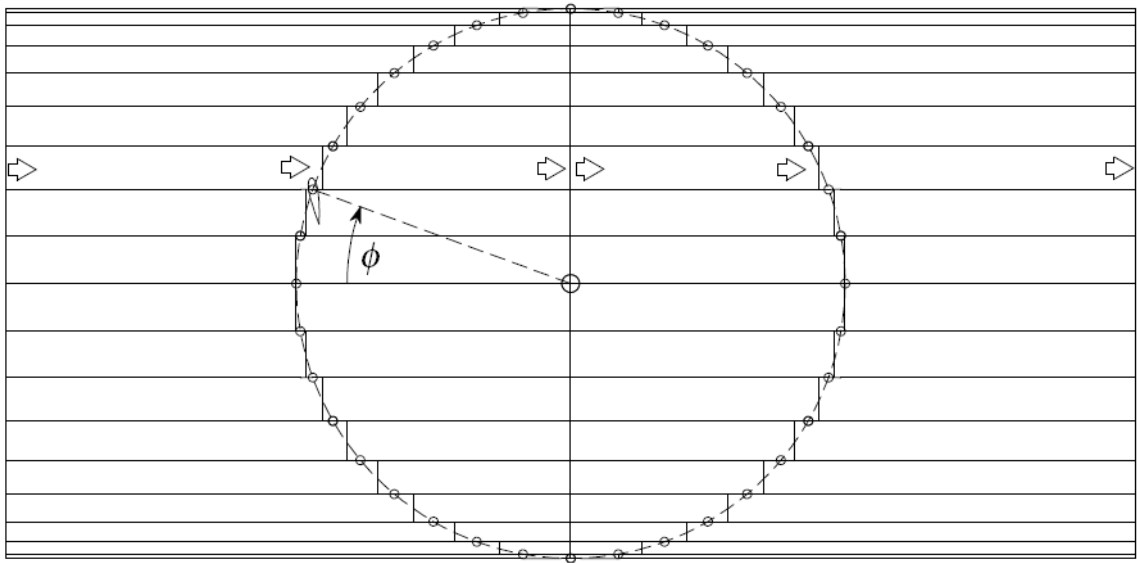
$$0 = f_{mom}^* - f_{aero}^* \quad (2.19)$$

The stream tube model can predict the performance of a wind turbine, but will typically predict higher power levels than those obtained from experimental results (Islam 2006). This type of a model suffers from the inability to accommodate the dynamic flow as the blade angle-of-attack changes during rotation and suffers performance variations at high TSR and solidity (Strickland 1975, Paraschivoiu 1981).

2.2.1.1 The Multiple Stream Tube

A remedy for the non-uniform flow across the tube was the use of a multiple stream tube model, as proposed by (Strickland 1975, Wilson 1974). This model divides the flow across a VAWT into multiple parallel tubes, shown in Figure 2-9. For each tube, an independent calculation is performed to find the momentum balance. Each tube is not necessarily of the same width. Tubes closer to the parallel wind direction, or negative ninety and positive ninety degrees can be larger than those that are close to the perpendicular to the wind region. The flow in a stream tube model is assumed to be uniform through each tube. The velocity solution for this model is similar to the momentum model with a single tube, except the velocity is solved for each tube.





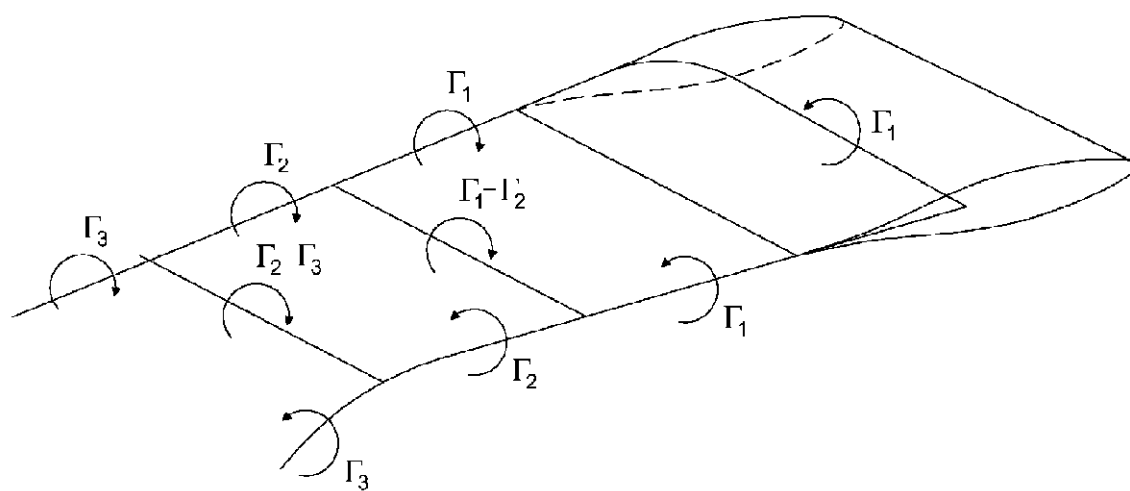
during rotation to optimize power. Both of these experiments show discrepancies in performance when compared with the results from the DMS model.

2.2.1.3 Momentum Models Summary

A momentum model of a VAWT using DMS can approximate the VAWT blade aerodynamic forces generated from the wind during rotation. Although momentum models break down at low TSRs, due to the equations sometimes being unsolvable, they can provide quick and adequate results for an initial VAWT design. The execution time for a solution to the momentum model is proportional to the number of stream tubes. In addition, this type of model could include modeling of circulation control on the blades by modifying the momentum balance solution and the lift and drag coefficients.

2.2.2 Vortex Model

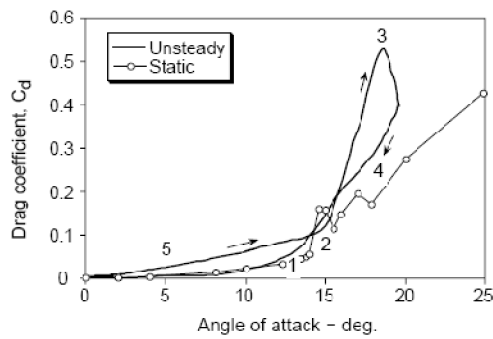
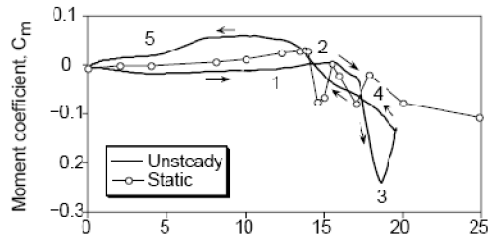
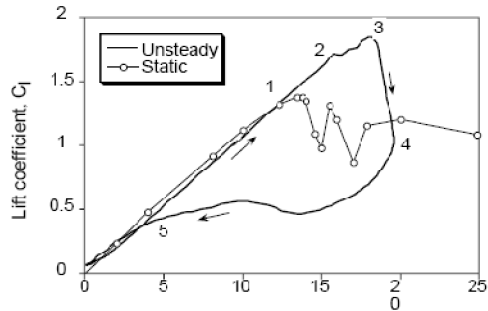
Vortex models are potential flow models that calculate velocities based upon vortices created in the wake of an airfoil. This method, unlike the momentum method, incorporates unsteady flow conditions, interaction between a blade and wake, finite aspect ratios, and flow curvature. Vortex models, either free or fixed, follow the conservation of circulation by modeling discrete shed vortices whose strength is equal to the change in bound circulation over time. Bernouilli's equation for incompressible flow can be used to evaluate blade lifting forces. A basic diagram of a vortex model is shown in Figure 2-11, where Γ_x is the circulation at that particular part of the flow. A grid of the circulation is developed, in two or three dimensions, of the area of the VAWT to be modeled. (Migliore 1978, Strickland 1980, Paraschivoiu 2002)



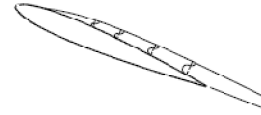
2.2.3 Dynamic Stall Modeling

Unlike HAWT, VAWT blades are subject to continually varying angles of attack and relative velocities even if the wind velocity is constant. These varying parameters do not fit well with static blade coefficients, therefore certain extensions need to be made for behavior such as stall.

Stall is when flow separation occurs and an airfoil suddenly no longer produces as much or any lift and an increase in drag occurs. This usually happens when an airfoil determined angle-of-attack has been exceeded. The exact angle at which stall occurs is dependent on the angle-of-attack rate of change. Typically airfoil lift and drag coefficients are gathered from measurements made while in steady state. Dynamic stall typically occurs at a different angle from the measured steady state angle. This change in the exact stall angle is termed dynamic stall. An overview of the dynamic stall process is shown in Figure 2-12. (Strickland 1983, Leishman 2002, Paraschivoiu 2002, Pawsey 2002, Larsen 2007)



Stage 1: Airfoil exceeds static stall angle, then flow reversals take place in boundary layer.



Stage 2: Flow separation at the leading-edge, followed by the formation of a 'spilled' vortex. Moment stall.



Stage 2-3: Vortex convects over chord, it induces extra lift and aft center of pressure movement.



Stage 3-4: Lift stall. After vortex reaches trailing-edge, the flow over upper surface becomes fully separated



Stage 5: When angle of attack becomes low enough, the flow reattaches to the airfoil, front to back.



$$\alpha_B$$

$$\gamma$$

$$\dot{\alpha}_B$$

$$\alpha_m = \alpha_B - \gamma K_1 \sqrt{\frac{c\dot{\alpha}}{2V}} \sin(\dot{\alpha}_B) \quad (2.20)$$

$$\gamma_L = 1.4 - 6 \left(0.06 - \frac{t}{c} \right) \quad (2.21)$$

$$\gamma_D = 1 - 2.5 \left(0.06 - \frac{t}{c} \right) \quad (2.22)$$

$$\gamma_M = \gamma_D \quad (2.23)$$

The drag and moment coefficients can be found in a lookup table from this angle-of-attack. The lift coefficient change is expressed by equation (2.24), where α_0 is the angle-of-attack where lift is zero.

$$c_L = c_L(\alpha_m) \left(\frac{\alpha_m}{\alpha_B - \alpha_0} \right) \quad (2.24)$$

2.2.3.2 The MIT Dynamic Stall Model

The MIT dynamic stall model is another dynamic stall model that was developed by (Johnson 1970) for application to helicopters. This model was later modified by (Noll 1982) to accommodate a VAWT. In this model, drag and lift are expressed as an exponential decay when dynamic stall is detected. Dynamic stall is predicted by evaluation of equation (2.25), whose parameters are similar to the Gormont method.

$$\alpha_{DS} = \alpha_{SS} + \gamma \sqrt{\left| \frac{c\dot{\alpha}}{2V} \right|} \quad (2.25)$$

To bridge the gap between static stall and dynamic stall, the coefficient of lift is modified by equation (2.26), where α_s is the slope of the static lift curve at static stall. When the dynamic stall angle has been reached equation (2.26) is used until the lift coefficient reaches a maximum, or the angle-of-attack begins to decrease. Once the maximum lift coefficient has been reached it begins to decay

according to equation (2.27) where $c_{L\max}$ is a predetermined maximum lift coefficient, ϑ is the current azimuth angle, and ϑ_0 is the azimuth angle where stall started.

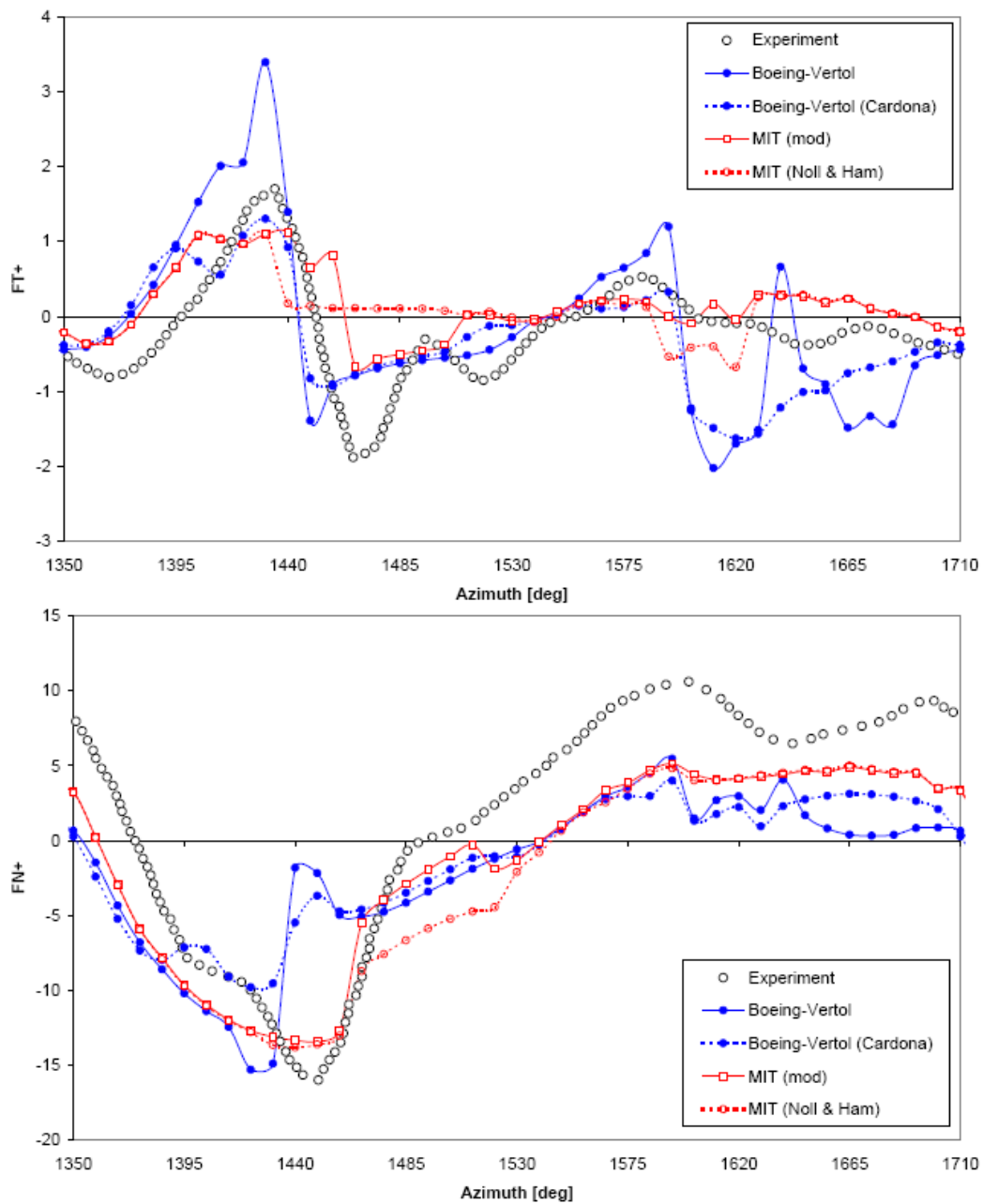
$$c_L = a_s \sin \alpha \quad (2.26)$$

$$c_L = \left[c_{L\max} - c_L(\alpha) \right] e^{-(\vartheta - \vartheta_0)2R/c} + c_L(\alpha) \quad (2.27)$$

2.2.3.3 Comparing Methods

To compare the dynamic stall methods, a 2-bladed VAWT was simulated using the MIT and Boeing-Vertol methods by (Pawsey 2002) and compared to an experimental VAWT, the non-dimensional tangential (F_T) and normal (F_N) forces are shown by Figure 2-13. The blades were a NACA0018 with a blade length of 0.61 m and turbine diameter 0.61m. These plots demonstrate how much dynamic stall can impact the forces generated by a VAWT. Several different methods were employed, each with different results. Since the models do not agree with each other, in terms of force estimation, choosing a dynamic stall to best fit the application may not be possible. (Pawsey 2002) suggests that the MIT method with modifications by (Noll 1982) is the most accurate model.

(Pawsey 2002) compared the coefficient of performance without and with a dynamic stall model. The findings were that the dynamic stall model predicted lower performance levels than without, which had a closer correlation to experimental work. The effects of dynamic stall for circulation control augmented blades currently remains unknown. Since dynamic stall lowers the performance predictions it is unnecessary to implement, when modeling a CC-VAWT, until a proper dynamic stall model can be created for an augmented blade.



prediction of performance, when compared to experimental results, in the momentum model can partially be attributed to the use of a constant uniform wind speed.

Wind turbine design is usually performed for a particular steady wind condition. Analysis is then later performed to determine ‘real-world’ performance results by using a wind model that incorporates disturbances. Wind disturbances can be modeled by a stochastic system characterized by a probability density. Turbulence has little effect on the long term power generation, which is dominated by the steady mean wind speed. Its effects can be seen on aerodynamic loads and the quality of the power. The two most accepted probability spectrum models are the von Karman and Kaimal spectrum. These methods incorporate a white noise function as the stochastic information source, along with several parameters that are determined experimentally for the location of the turbulence model. (Veers 1984, Paraschivoiu 2002, Bianchi 2007)

The wind probability distribution for a given average wind can be found using a Weibull distribution as defined by

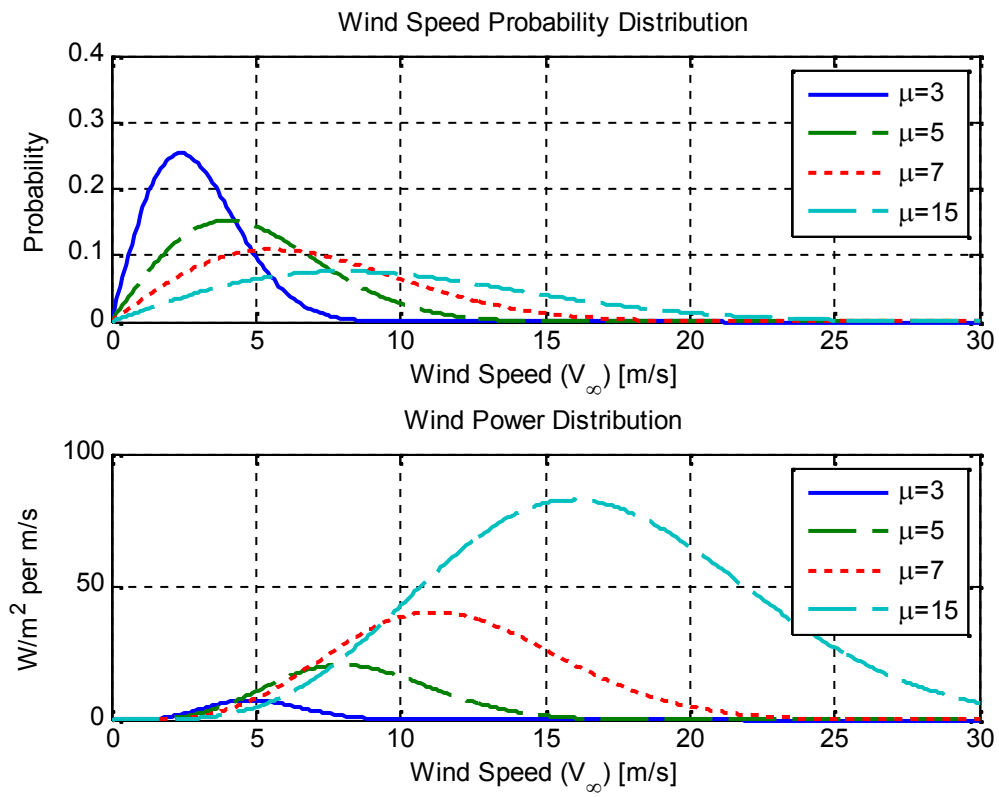
$$f_w(x; \lambda, k) = \frac{k}{\lambda} \left(\frac{x}{\lambda} \right)^{k-1} e^{-\left(\frac{x}{\lambda} \right)^k} \quad (2.28)$$

Using a Weibull distribution with a shape parameter (k) of $\frac{2\mu}{\sqrt{\pi}}$ and μ being the average wind speed, a Rayleigh distribution can be generated. This distribution determines the probability of a wind speed for a given average wind speed. Typically an installation site for wind turbines is evaluated and an average wind speed is collected. A Rayleigh distribution can then be defined as a function of the Weibull distribution or

$$f_R(x; \mu) = f_w \left(x, 2, \frac{2\mu}{\sqrt{\pi}} \right) \quad (2.29)$$

$$P = \frac{1}{2} \rho_{\infty} f_R(V, \mu) V^3$$

$$\frac{kg}{m^3}$$



2.2.5 VAWT Aerodynamic Modeling Summary

The two major methods as described earlier for modeling a VAWT are the momentum and vortex models. The less accurate and less computationally complex method is the momentum model. On the other hand, the vortex model method can be highly accurate but computationally complex. These models can be used during the design of a VAWT for their strengths to produce quick results (momentum) or high accuracy (vortex model). In addition, dynamic stall and wind speed probability distributions can be modeled to provide behavior predictions. These aerodynamic models, when implemented, can allow a designer to adjust parameters such as solidity, tip speed ratio, airfoils geometry, turbine height and/or diameter, and the circulation control conditions to examine their effect on performance.

2.3 Circulation Control

Circulation control can increase the bound circulation of an airfoil, and in turn produce more lift. This is typically implemented by a tangential slot of a high-velocity jet blowing over a rounded surface. The slotted surface can be the leading, trailing, top, or bottom edges of an airfoil. The blowing causes the boundary layer and the jet sheet to remain attached along the curved area of the airfoil due to the Coanda effect, which causes the jet to flow without separation. If a trailing edge slot is used, lift can be substantially increased. (Abramson 1975, Angle 2008)

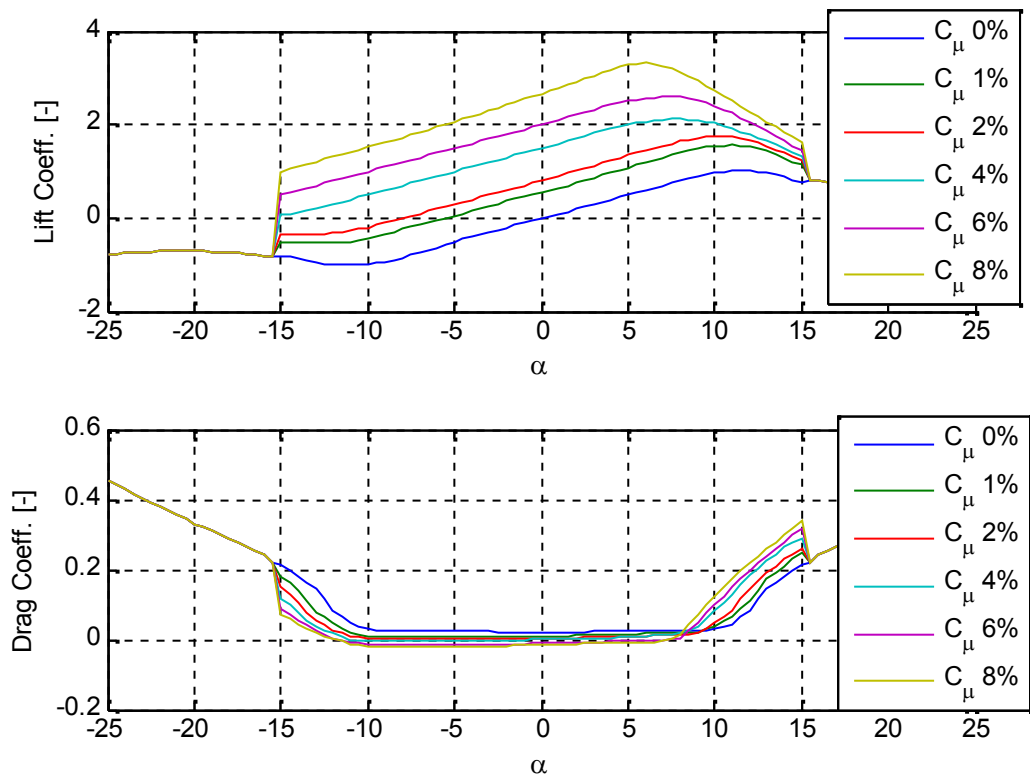
The use of circulation control enhances the lift produced by an airfoil when used properly. An experimental airfoil with blowing slots was tested by (Loth 1976) and a relationship between the jet velocity, jet height, blade chord, and free stream velocity was found. This can be represented by equation (2.31) which demonstrates the additional lift as determined by (Loth 1976) where the constants in the equation were most likely attributed to the blade shape. The forward velocity (V_∞) and

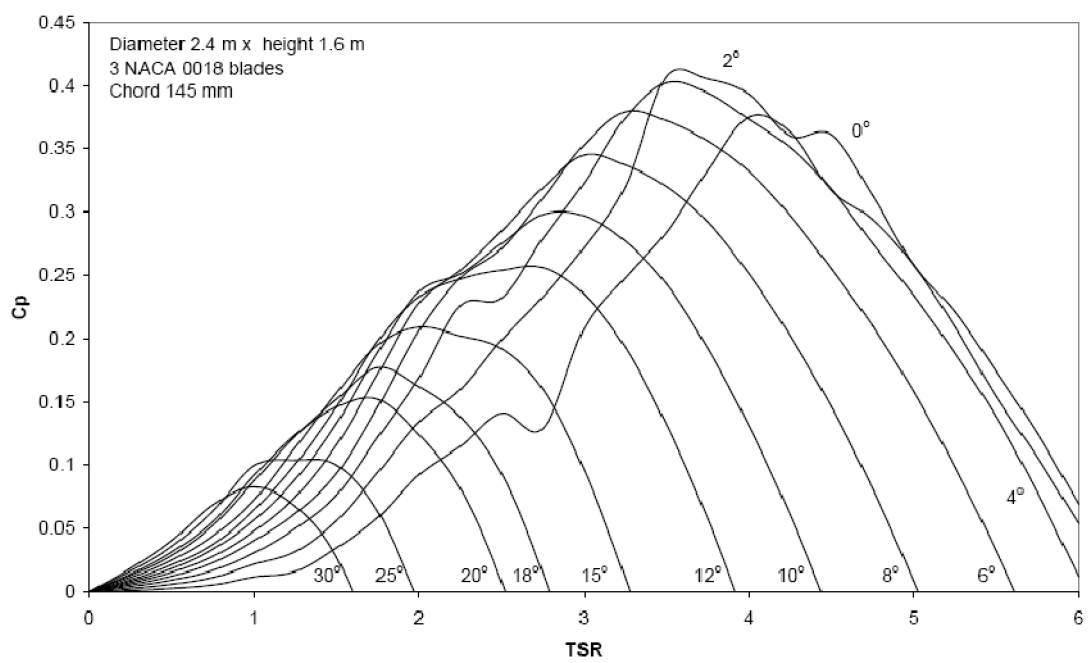
blowing slot jet velocity (V_j) were varied to determine a relationship for the added lift. The span-wise coefficient of blowing, used in circulation control evaluations, is defined by equation (2.32).

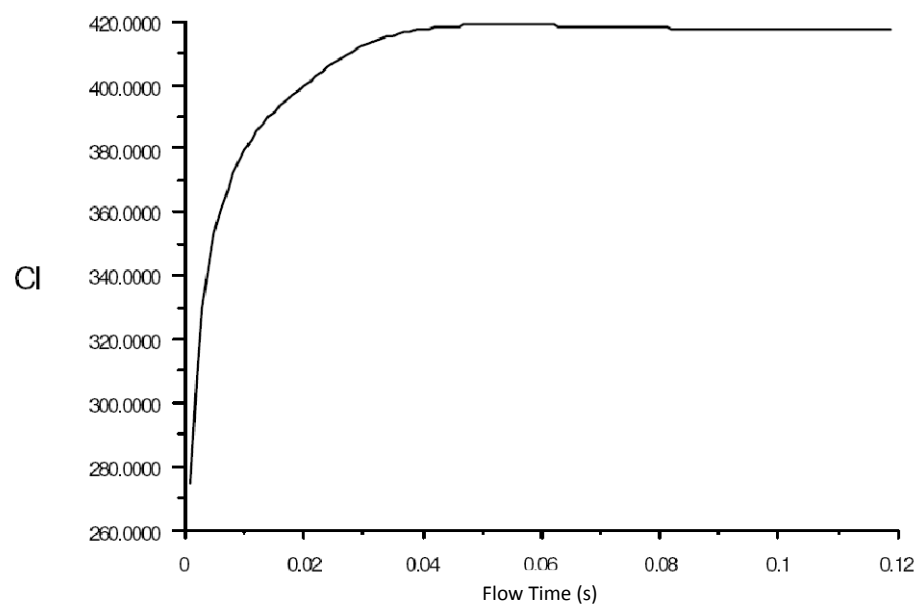
$$\Delta C_L = 40 \left(\frac{h}{c} \right)^{0.64} \left(\frac{V_j}{V_\infty} - 1 \right) \quad (2.31)$$

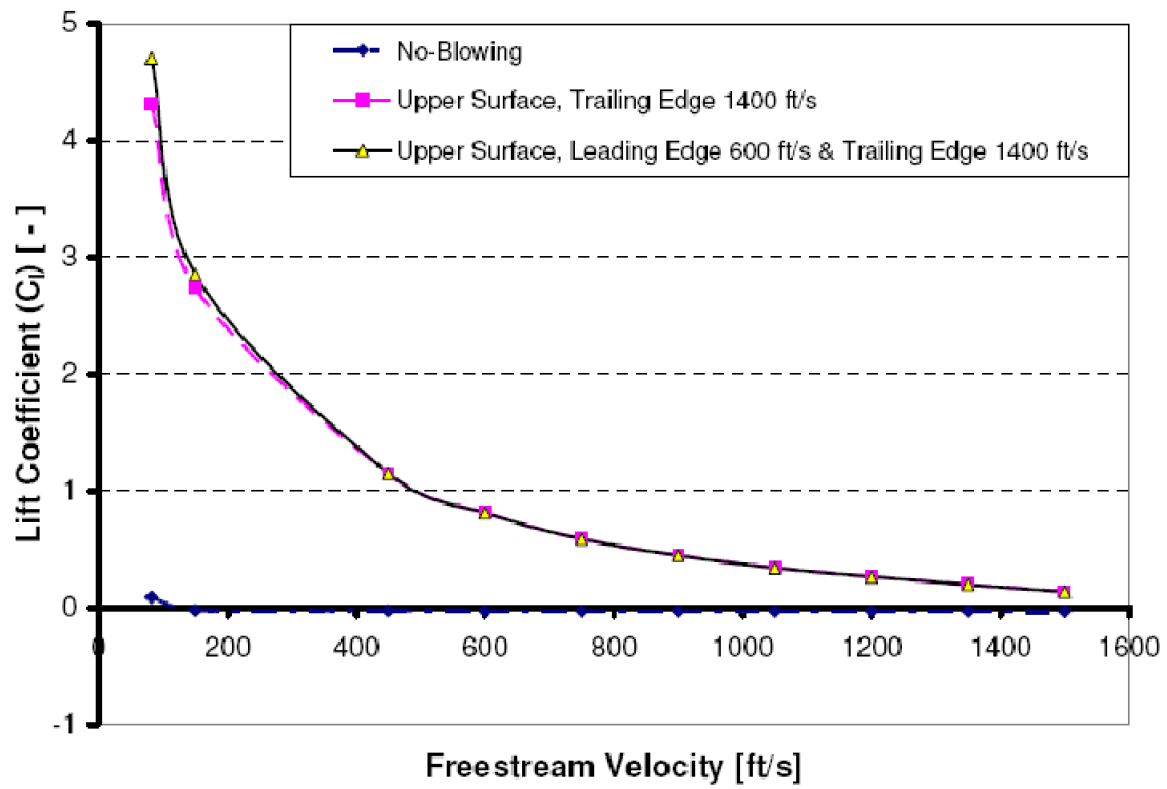
$$C_\mu = \frac{2hV_{jet}^2}{CV_\infty^2} \quad (2.32)$$

Elliptical airfoils are commonly used for circulation control augmentation due to their ratio of lift enhancement to the amount of blowing. From the literature, lift and drag coefficients for various blowing coefficients of elliptical airfoils were found and compiled. The elliptical blade performance and the effects of circulation control on lift and drag can be seen in Figure 2-15. Lift and drag coefficients were only available for angles-of-attack within ± 15 degrees, coefficients outside of this region were assumed to be the same as the no-blowing case which has been tested for larger angles-of-attack. The airfoil coefficients of lift and drag for this elliptical airfoil with circulation control represents the blade performance which be used in a analytical model of a VAWT.(Abramson 1975, Englar 1972, Wood 1981, Abramson 1984, Rodman 1986, Franke 1988, Novak 1987, Abramson 2004, Alexander 2005, Wetzel 2009)









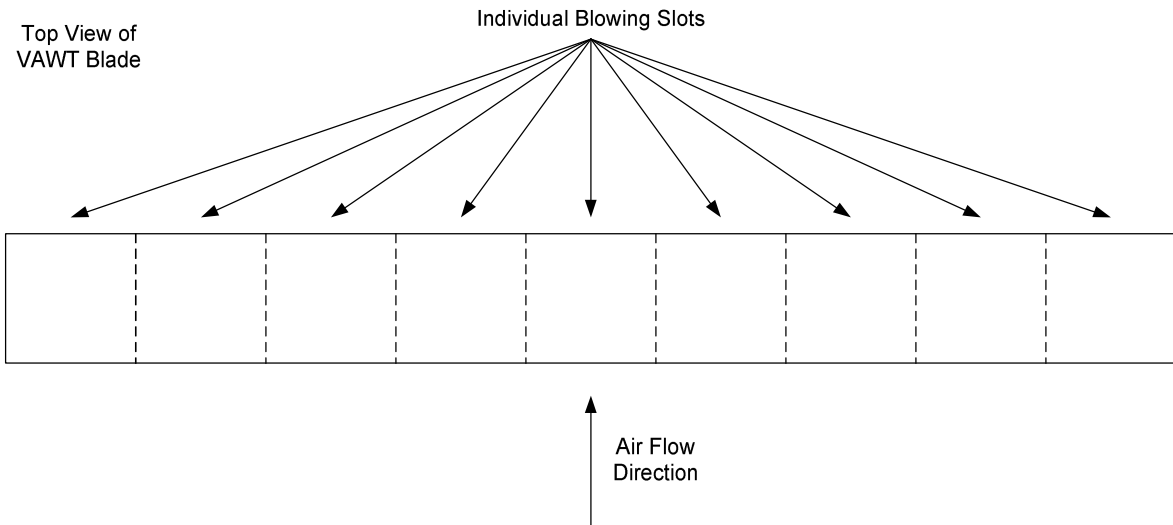


Figure 2-19 Possible Blade Blowing Slot Configuration

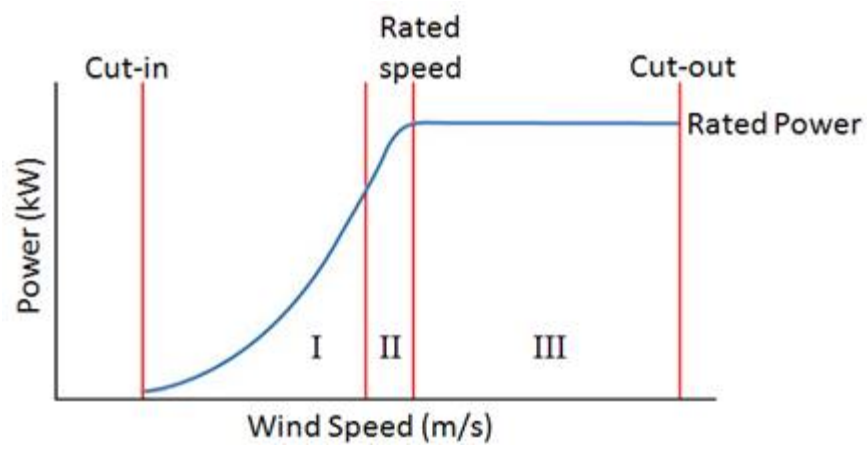
2.4.2 VAWT Blade Summary

Circulation control on a VAWT blade with leading, trailing, upper, and lower multiple independently controlled slots could allow lift to be controlled for a variety of applications. This could be similar to a variable angle-of-attack blade without physically changing the angle-of-attack. Also, a blowing slot configuration may exist that will reduce lift and increase drag to effectively slow down rotation in an over-speed situation.

2.5 Controlling Wind Turbines

Wind turbine control, in general, is achieved by varying aerodynamic properties of the turbine to effect torque production, vary the speed, or more commonly by varying the load during rotation. The goal of a wind turbine controller is to maximize power output, minimize stress or fatigue loads, limit the turbine from excessive rotational speeds, and/or start the rotation of the turbine. Various techniques of control designs exist for HAWT and VAWT turbines.

The ideal case for wind turbine control is when the power output of a turbine reaches a design or rated wind speed. Once this speed has been met, the power output is at its maximum and the turbine



startup, and overall performance. These types of VAWTs can have blades that are either independently or collectively pitch controlled.

(Brulle 1977, Drees 1978, McConnell 1979) each proposed slightly different methods of active, variable pitch controlled VAWTs to control performance. These early turbines showed that large pitch changes at low TSR produces good performance and small pitch changes at low TSR do not perform as well as high TSRs. No attempt at optimization, maximum power level seeking, or adapting to changing wind conditions were attempted. (Kirke 1998)

Free pivot blades have their angle-of-attack determined by the pitching moment produced by aerodynamic and kinematic forces. (Kirke 1998, Pawsey 2002) developed, simulated, and tested free pivot VAWT. A large unknown quantity of these types of turbines is the actual pitch of the blades. The momentum and vortex models were both used to estimate how the blade would react. (Pawsey 2002) was only able to measure blade pitch angles for a small range of TSRs.

(Sedighzadeh 2008) developed an adaptive reinforcement learning Proportional Integral Derivative (PID) controller for rotational speed control of a wind turbine. In this case, wind disturbances were compensated by use of an adaptive controller. Results show that the adaptive controller outperformed a non-adaptive PID controller. This was most likely due to the ability to better track wind disturbances.

Actively controlled variable pitch blades should provide insight into methodologies for the selection of the appropriate angle-of-attack. This angle-of-attack could be compared to the effect that circulation control produces.

2.5.3 Speed and Torque Control

Wind turbines exist that are controlled to rotate at a constant speed. However, improved performance can be achieved if the rotational speed can be made to vary by matching or tracking the maximum performance point at the current wind speed. (McIver 1996) demonstrated a 10% improvement of a variable speed over a constant speed HAWT. The goal of the controller was to maintain a constant TSR. The operating point TSR was determined from the relationship between TSR and the coefficient of performance.

Three methods of controlling wind turbines are suggested by (Bianchi 2007), speed and torque control, and a hybrid of the two. The speed controller typically uses blade pitch to drive the turbine to a reference rotational speed. This type of a controller can be useful for turbines that must produce a constant frequency AC voltage. This is shown by Figure 2-21 where V_{f-est} is the estimated forward wind velocity, ω_{ref} is the reference speed, ε is the error of the reference speed to the actual speed, θ is the rotational position, V_f is the actual forward wind velocity, and ω is the rotational speed.

A torque controlled turbine controller will drive the turbine to always produce a specified torque, typically by changing blade pitch. This type of a controller can be more efficient, due to the focus on maximum torque generation and not rotational speed, as is shown by Figure 2-22. Using a torque controller may require the use of a transmission on the output shaft of the turbine to provide, if necessary, constant speed for a generator. A hybrid controller of the speed and torque controller also exists to further the efficiency of the turbine, by combining the torque and speed to maximize power, shown by Figure 2-23.

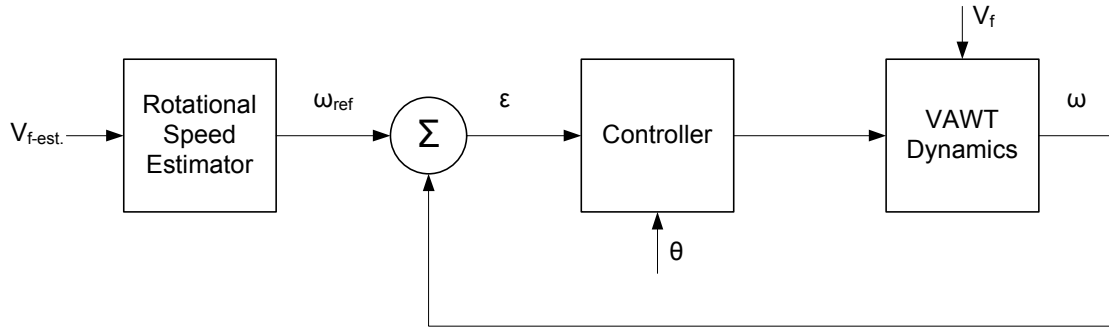


Figure 2-21 Speed Controller (Bianchi 2007)

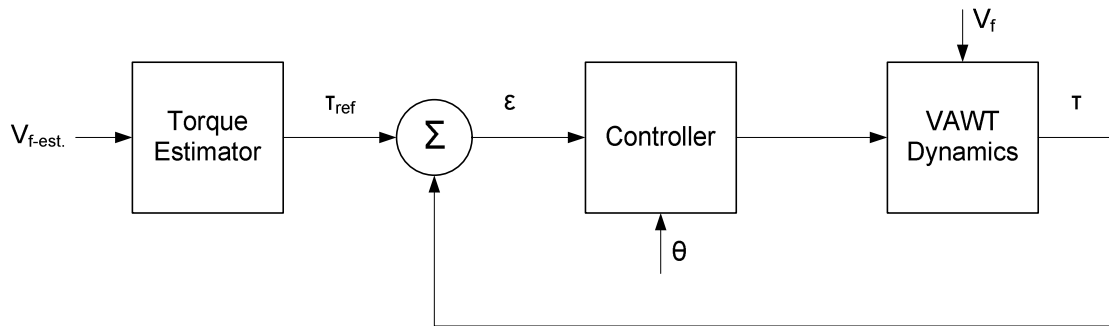


Figure 2-22 Torque Controller (Bianchi 2007)

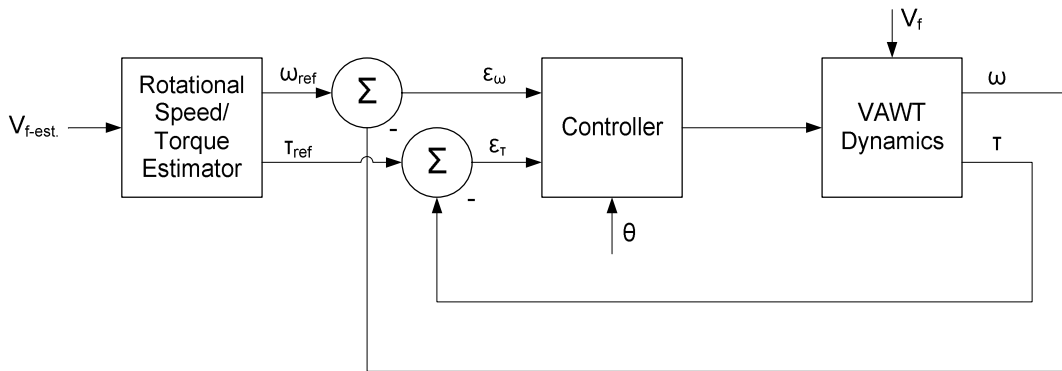


Figure 2-23 Hybrid Speed/Torque Controller (Bianchi 2007)

These controllers presented by (Bianchi 2007) are a dual-controller that is gain scheduled. This means that there are two separate controllers that have a scheduler to determine which controller to use. This methodology allows one controller to run the startup sequence, and another controller to maintain steady state. Although this controller has two sub-controllers and a scheduler, it is possible to have many more controllers under the command of a scheduler.

Variable pitch VAWT blades have been experimentally tested to improve power generation. (Staelens 2003) proposed a straight bladed variable pitch VAWT. Three different pitch control mechanisms were used to increase forces developed by the blades. The first method was to keep the blade out of stall by changing the angle-of-attack during rotation as shown by $\Delta\alpha_e$ in Figure 2-24. The second method was to improve upon the jump required in angle-of-attack control as the azimuth crosses negative ninety and positive ninety degrees as shown by $\Delta\alpha_e$ in Figure 2-25. The third method is a sinusoidal function of the difference between the geometric angle-of-attack and the stall angle and was created to provide a continuous angle-of-attack as shown by $\Delta\alpha_e$ in Figure 2-26.

A significant increase in power was found by using these three methods, shown in Figure 2-27. Although the first method provides the best performance, it is “physically and mechanically impossible to realize(Staelens 2003).” The second method produces less power and contains a discontinuous angle-of-attack region which can lead to early fatigue of the blades. The third method which produces still lower power than the second method was the only method that “[had] the advantage of being practically feasible.”

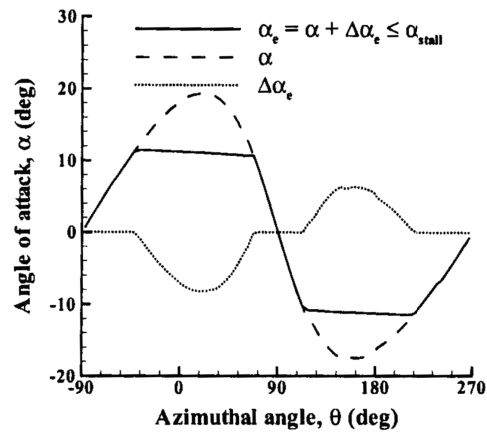
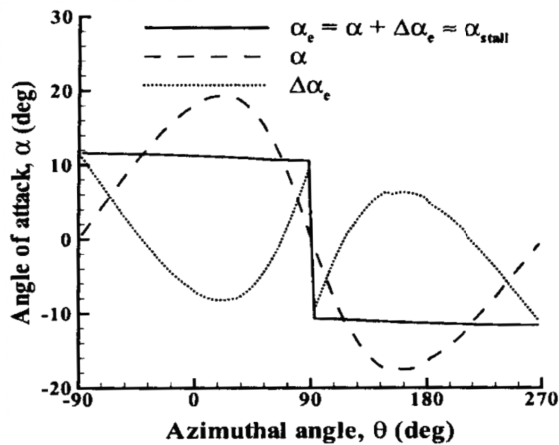


Figure 2-24 Pitch Control Mod I (Staelens 2003)

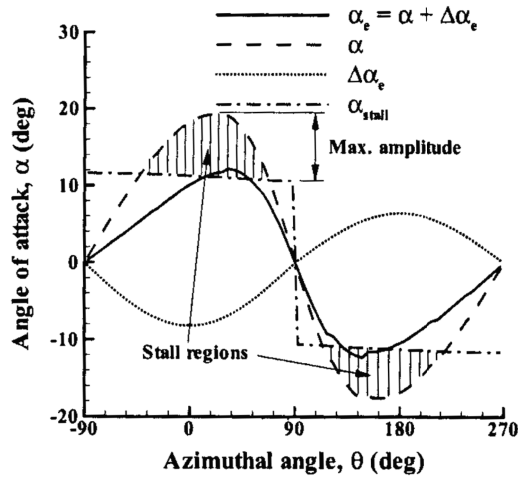


Figure 2-26 Pitch Control Mod III (Staelens 2003)

Figure 2-25 Pitch Control Mod II (Staelens 2003)

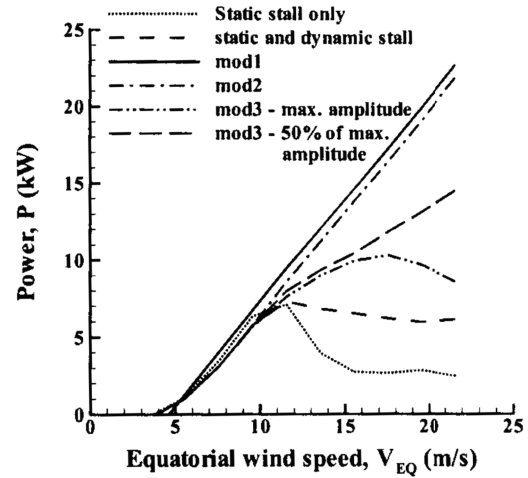


Figure 2-27 Pitch Control Power (Staelens 2003)

2.5.4 Controlling the CC-VAWT

A method of controlling the CC-VAWT for maximum power generation would be to follow the techniques of the pitch controlled turbines and map the blowing slot parameter C_μ to the angle-of-attack. This would allow the CC-VAWT to perform similarly to a pitch controlled VAWT such as the work performed by (Staelens 2003). Startup and low TSRs could allow a controller to dynamically vary the blowing slots to maximize a constant rotational force. This could allow for startup at speeds lower than possible for previous VAWTs. Once steady-state rotation has been achieved, the angle-of-attack of the blades varies little for moderate TSRs. At this point, a controller can seek the maximum power achievable and control the blowing slots to meet this. In addition, individual control over each blade at many positions during rotation may occur. A controller could select the proper blowing parameters to seek maximum power based upon rotation speed and torque.

2.5.5 Controlling Wind Turbines Summary

Various control methodologies have been reviewed for controlling a CC-VAWT. Speed and torque control of wind turbines to develop the most capable power can be achieved by varying the pitch

of the blade according to pre-determined tables or gain scheduling. A controller could handle varying wind conditions, develop start up rotation, maximize the power output, minimize the blade stress, and schedule blowing slots individually on a CC-VAWT.

2.6 Literature Review Summary

Aerodynamics, modeling, and controls that could be applied to a CC-VAWT have been reviewed. Basic aerodynamics of the VAWT and its blades provide an insight to how the machine functions. Analytical models such as the momentum and vortex methods can provide forces generated by the blades and the generated power to help in the system design. Scheduling of the circulation control for a VAWT, which has never been attempted before, for maximum energy capture could be accomplished by analyzing how the power is produced during rotation by adjusting the VAWT parameters. Solidity, TSR, blade characteristics, and rotational speed may all be adjusted and impacted by circulation control.

Chapter 3. CC-VAWT Analytical Modeling

3.1 Introduction

Two fundamentally different aerodynamic modeling methods exist for VAWT which were covered in sections 2.2.1 and 2.2.2 of the literature review. First, the momentum method, based upon the conservation of momentum principle, was implemented for preliminary performance information. Second, the vortex model, implemented in both 2D and 3D, also provided performance results and was used to develop a control scheme. These two models can agree on performance predictions and have numerical solution divergence issues. The effects of circulation control were included in both models.

Equation Chapter 3 Section 1

3.2 The Double Multiple Stream Tube Model

The double multiple stream tube is the most advanced method of momentum modeling for a VAWT. Multiple stream tubes are used and the turbine is divided into an upwind and downwind section. The multiple stream tubes allow for a non-steady relative blade velocity and the upwind/downwind sections allow for the energy taken out of the upwind half to be taken into account downwind.

The upwind half of the tube is located fore of the center of rotation, and the downwind half is located aft of the center of rotation, shown by Figure 3-1. This model carries out a momentum balance in each half of every tube. This method of modeling enhances the multiple stream tube model to account for high TSRs where the downwind half of the tube has a significantly lower flow than the upwind tube. The velocity solution procedure is similar to the multiple stream tube, except the solution for the upwind velocity is fed into the downwind half of the model.

The sign convention and force direction used for this model are shown in Figure 3-1. The arrows protruding from the orbits of varying radii represent the magnitude and direction of the relative blade

velocity at various TSR's (λ) and orbital positions (θ). The angle of attack (α), resultant lift force (L), resultant drag force (D), axial (A) and normal (N) forces, and the wind speed (V_∞) are included in the figure. To complete a simulation, the upwind induction factor is first found and then used in the downwind solution.

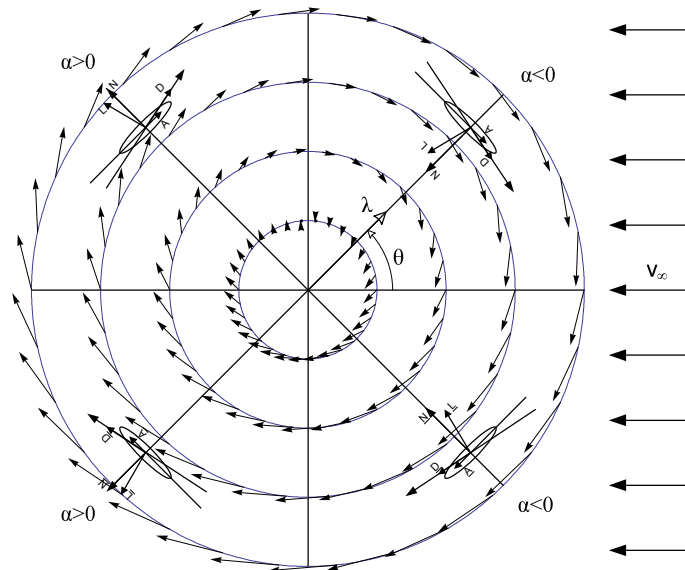


Figure 3-1 Relative blade velocities and angles of attack at various tip speed ratios λ and blade positions θ

3.2.1 Upstream Solution

Since energy is extracted from the flow, a pressure gradient exists across the turbine capture area, similar to that of a Froude actuator disk. The velocity of air impinging upon the turbine, v_1 , is somewhat less than the free stream velocity, v_∞ , since the blades offer a partial obstruction to the flow causing a pressure to build up in front of the turbine. The locations across the turbine where the velocity varies are shown in Figure 3-2. Once the air moves past the blades of the turbine, the velocity is further reduced to v_2 as the flow momentum is reduced. The initial reduction in velocity to v_1 can be mathematically modeled by an induction factor, a , by the commonly used equations (3.1) and (3.2) from equation (3.3).

$$v_1 = (1 - a)v_\infty \quad (3.1)$$

$$v_2 = (1 - 2a)v_\infty \quad (3.2)$$

$$v_1 = \frac{1}{2}(v_\infty + v_2) \quad (3.3)$$

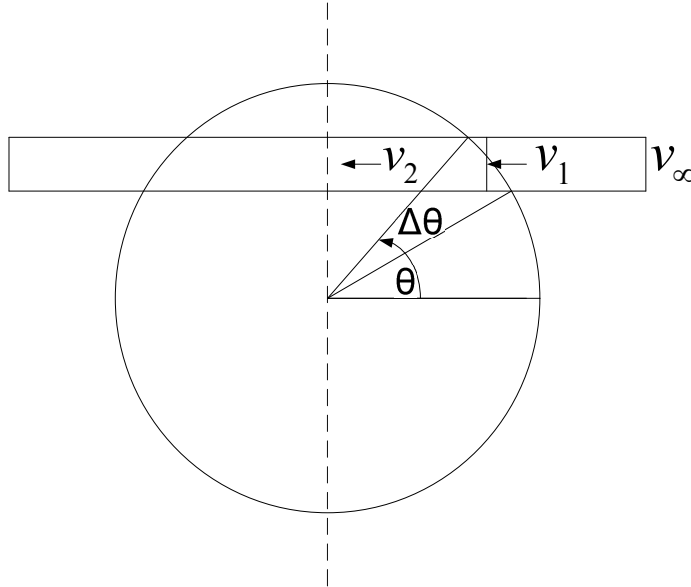


Figure 3-2 Multiple Stream Tube Velocity Boundaries

An equation for the average force, f_{ave} , exerted on the area of a stream tube, due to the loss of momentum of the air when slowing down from v_1 to v_2 , can be derived from momentum theory. Force, f , is the time derivative of momentum, mass, m , times a change in velocity, Δv , as given by equation (3.4) where the dot notation indicates a time derivative.

$$f = \frac{d}{dt}(m \cdot \Delta v) = \dot{m} \cdot \Delta v + m \cdot \Delta \dot{v} \quad (3.4)$$

In steady state flow, the acceleration, or time derivative of velocity, is zero and can be expressed in equation (3.5) for the force exerted due to a change in velocity of Δv , where the mass flow is given in terms of area, A , the density, ρ , and the flow velocity, v .

$$f = \dot{m} \Delta v = \rho \cdot A \cdot v \cdot \Delta v \quad (3.5)$$

The case of the stream tube, this velocity change is $v_\infty - v_2$ resulting in equation (3.6).

$$f_{ave} = \rho \cdot A \cdot v_1 \cdot (v_\infty - v_2) \quad (3.6)$$

Substituting for v_1 and v_2 in terms of v_∞ and the induction factor a , this can be re-expressed in equation (3.7) for the average force exerted in the stream tube.

$$f_{ave} = \rho \cdot A \cdot (1-a) \cdot 2a \cdot v_\infty^2 \quad (3.7)$$

The area, A , of the stream tube is the width, $r \cos(\theta) \Delta\theta$, multiplied by its height, h , resulting in equation (3.8).

$$f_{ave} = 2 \cdot \rho \cdot h \cdot r \cdot \cos(\theta) \cdot \Delta\theta \cdot (1-a) \cdot a \cdot v_\infty^2 \quad (3.8)$$

This is the average force exerted on the blades due to the loss in momentum of the flow, through a full 360 degree revolution. The actual force exerted on the blades can be found by undoing the averaging. The blades will spend time, equation (3.9) in a small portion of $\Delta\theta$ as they sweep out the semicircle of π radians, where N is the number of stream tubes.

$$t_{dwell} = N \frac{\Delta\theta}{\pi} \quad (3.9)$$

During this time, the blades receive the time average force, so the actual force, f_{mom} , during the period, t_{dwell} , has to be the average force divided by this time interval as given by equation (3.10).

$$f_{mom} = \frac{2 \cdot \pi \cdot \rho \cdot h \cdot r \cdot \cos(\theta) \cdot (1-a) \cdot a \cdot v_\infty^2}{N} \quad (3.10)$$

As an analytical convenience, this force in the direction of the flow can be non-dimensionalized in equation (3.11).

$$f_{mom}^* = \frac{N \cdot f_{mom}}{\frac{1}{2} \rho \cdot h \cdot r \cdot v_\infty^2} \quad (3.11)$$

This is valid for $a < 0.4$, where the wake is not turbulent. The * indicates that the force is non-dimensional. The wake becomes turbulent around $a > 0.4$, at which point an empirical formula, Glauert's

formula (Pawsey 2002), is more appropriate. The corresponding non-dimensionalized force can be expressed as equation (3.12).

$$f_{gl}^* = \pi \cos(\theta) \cdot \frac{26 \cdot a + 4}{15} \quad (3.12)$$

The overall non-dimensional thrust force exerted on the turbine can then be determined by equation (3.13).

$$f_{mom}^* = \begin{cases} \pi \cos(\theta) 4(1-a)a & a < 0.4 \\ \pi \cos(\theta) \left(\frac{26a+4}{15} \right) & a \geq 0.4 \end{cases} \quad (3.13)$$

The above force equation requires knowledge of the blade area, A_{bl} , the angle of attack, α , and the flow velocity, v_{bl} , as seen by the blade. The blade area is given by blade chord, c , multiplied by blade span, s . It is, however, more convenient to work with wind turbine related non-dimensional parameters, the tip speed ratio, equation (3.14), and the solidity factor, equation (3.15), where ω is the angular velocity of the turbine, r is the turbine radius and N is the number of blades.

$$\lambda = \frac{\omega \cdot r}{v_{\infty}} \quad (3.14)$$

$$\sigma = \frac{N \cdot c}{r} \quad (3.15)$$

The angle of attack resulting from the vector sum of the velocity induced due to blade motion, ω , r , and the impinging wind with velocity v_1 is given by equation (3.16).

$$\alpha = \tan^{-1} \left(\frac{\cos \theta}{\sin \theta - \frac{\lambda}{1-a}} \right) \quad (3.16)$$

The magnitude of the relative blade velocity, V_{rel} , is given by equation (3.17), which can be non-dimensionalized by with v_∞ resulting in equation (3.18).

$$V_{rel} = \sqrt{[v_1 \cdot \cos(\theta)]^2 + [v_1 \cdot \sin(\theta) - \omega \cdot r]^2} \quad (3.17)$$

$$V_{rel}^* = \frac{V_{rel}}{V_\infty} = \sqrt{(1-a)^2 + \lambda^2 - 2(1-a)\lambda \sin(\theta)} \quad (3.18)$$

The aerodynamic force per unit span is defined as a function of its lift and drag coefficients, relative velocity, turbine solidity, and angle of attack in the rotational direction can then be represented in equation (3.19).

$$f_{trq}^* = V_{rel}^2 \sigma [C_L \sin(\alpha) - C_D \cos(\alpha)] \quad (3.19)$$

The aerodynamic force per unit span in the direction of the wind as a function of its lift and drag coefficients, relative velocity, turbine solidity, blade rotational position, and angle of attack can be represented in equation (3.20).

$$f_{aero}^* = V_{rel}^2 \sigma [C_L \cos(\alpha + \theta) + C_D \sin(\alpha + \theta)] \quad (3.20)$$

3.2.2 Downstream Solution

The downstream multiple stream tube model, similar to the upstream model, can be represented by the use of another induction factor, b . A single downstream tube with upwind velocity is shown by Figure 3-3. This downwind factor can be included in the stream tube equations similar to the upstream method, where the upstream induction factor, a , can be included.

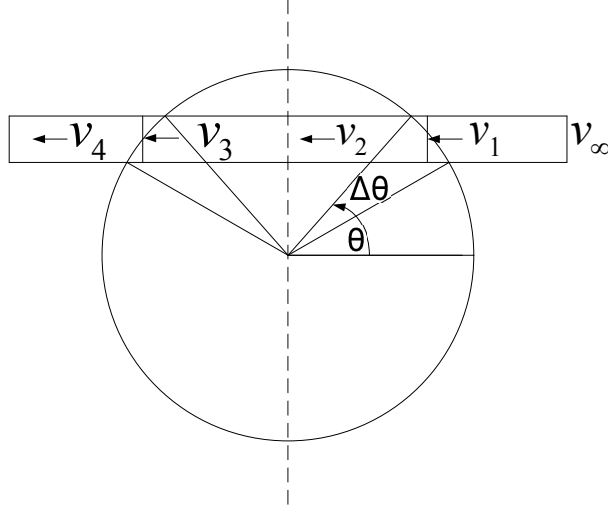


Figure 3-3 Double Multiple Stream Tube Velocity Boundaries

The initial reduction in velocity to v_3 can be mathematically modeled by an induction factor as expressed by equation (3.21).

$$v_3 = (1-b)v_2 = (1-b)(1-2a)v_\infty \quad (3.21)$$

The downwind blade angle-of-attack can be expressed in equation (3.22) and the blade relative velocity in equation (3.23) where c is represented by equation (3.24).

$$\alpha = \tan^{-1} \left(\frac{\cos(\theta)}{\sin(\theta) - \frac{\lambda}{(1-2a)(1-b)}} \right) \quad (3.22)$$

$$V_{rel}^* = \sqrt{(c^2 + \lambda^2 - 2c\lambda \sin(\theta))} \quad (3.23)$$

$$c = (1-b)(1-2a) \quad (3.24)$$

The momentum force downwind, incorporating the upwind wind induction factor, can be represented in equation (3.25). The aerodynamic forces in the rotational direction and in the wind direction are calculated the same as the upwind model.

$$f_{mom}^* = \begin{cases} \pi \cos(\theta) 4(1-b)(1-2a)b & b < 0.4 \\ \pi \cos(\theta) \left(\frac{(26b+4)(1-2a)}{15} \right) & b \geq 0.4 \end{cases} \quad (3.25)$$

3.2.3 Finding the Induction Factor

Identifying the induction factor for each stream tube is accomplished by solving for the induction factor where the two momentum and aerodynamic forces are equal but opposite (Leishman 2006, Spera 1994, Dreier 2007, Paraschivoiu 1986). A numerical solver algorithm is used to find the zero of equation (3.26). The non-dimensional momentum and aerodynamic force equations are non-linear and conventional analytical solutions are difficult to find for all parameter configurations. To solve this, first the upwind half ($-\pi/2$ to $\pi/2$) of the turbine induction factor, a , is found. Next, the induction factor is fed into the downwind stream tubes to solve for the downwind induction factor, b . Once these induction factors have been solved, the coefficient of performance can be calculated.

$$f_{mom}^* - f_{aero}^* = 0 \quad (3.26)$$

3.2.4 Coefficient of Performance

The induction factor, a , can be interpreted as the fraction by which the free stream velocity has been reduced before reaching the turbine blade. Using the solidity and tip speed ratio, the average airfoil forces can be computed. Non-dimensional power “density” per blade, p^* , can be defined from power, the product of torque, τ , and angular velocity, ω . This product is given equations (3.27) or (3.28).

$$p = \tau \cdot \omega = \left(n f_{aer} \frac{\frac{1}{2} \rho \cdot h \cdot r \cdot v_{\infty}^2}{N} \cdot r \right) \cdot \left(\frac{v_{\infty} \cdot \lambda}{r} \right) \quad (3.27)$$

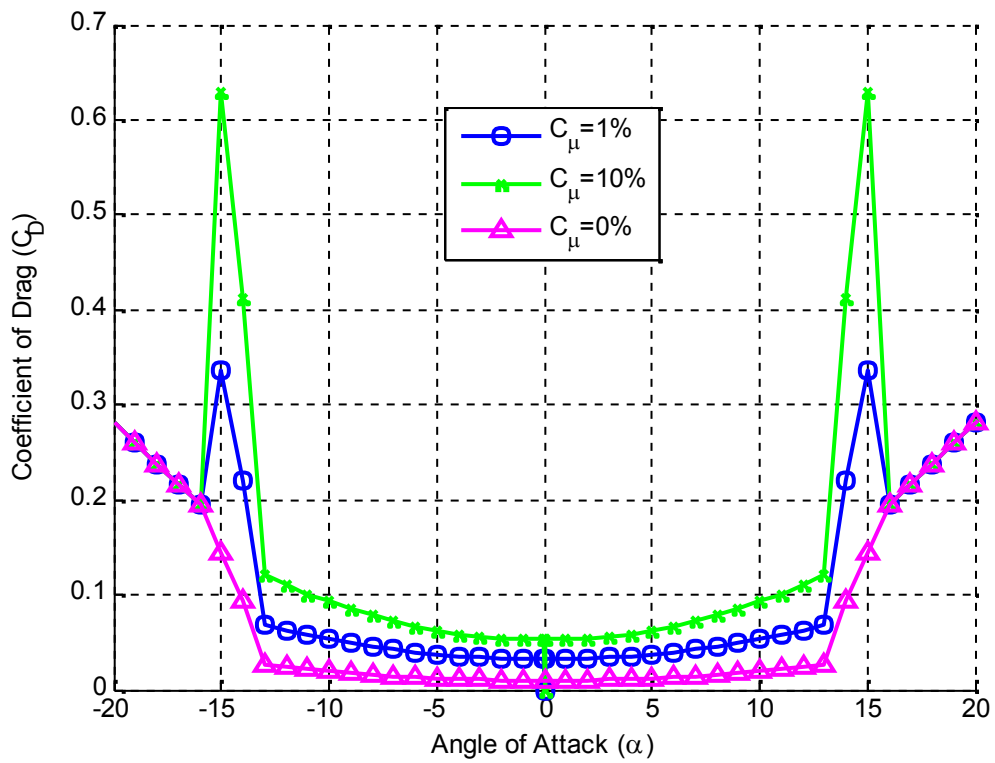
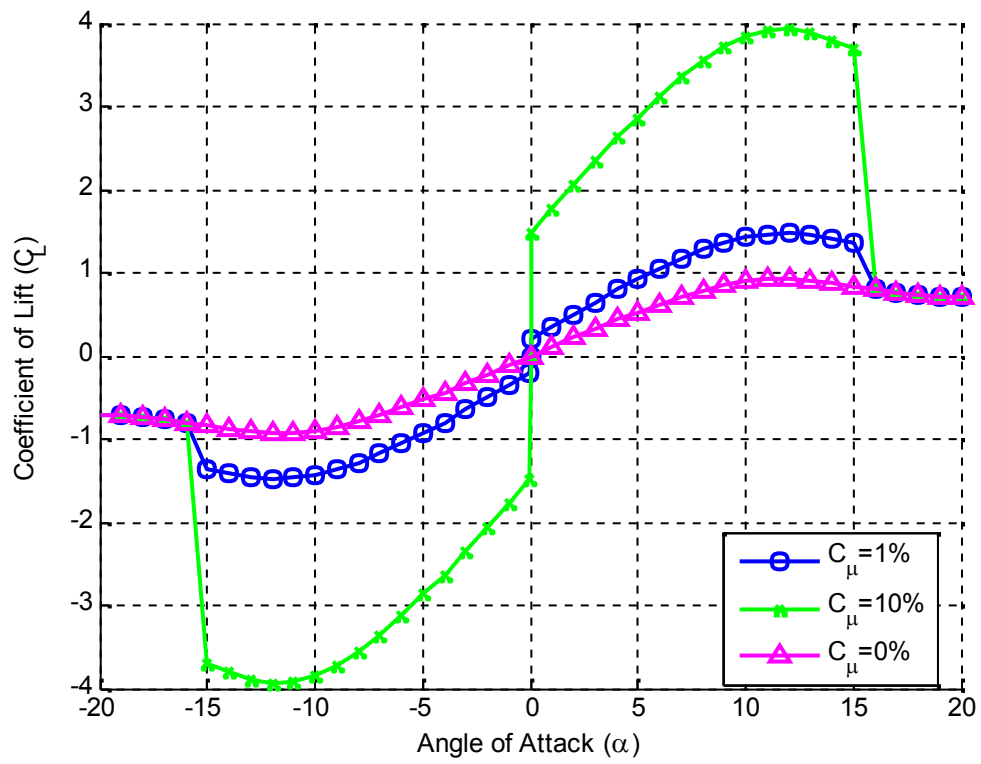
$$p^* = \frac{p \cdot N}{\frac{1}{2} \rho \cdot h \cdot r \cdot v_{\infty}^3} = \lambda \cdot f_{aer}^* \quad (3.28)$$

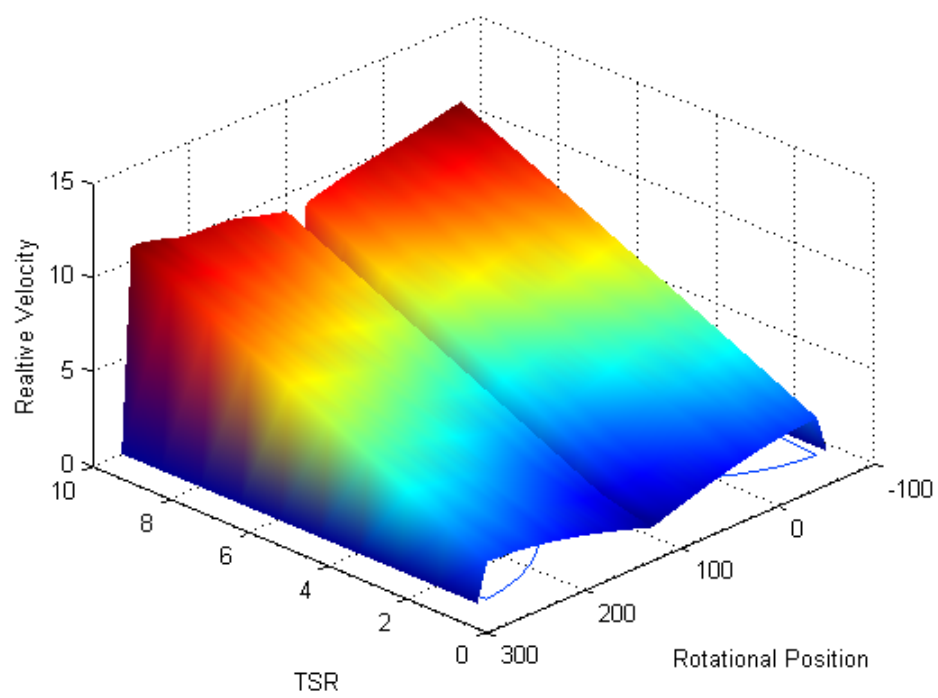
From this non-dimensional power density , p^* , the turbine's coefficient of performance, C_p , can be found through averaging over all blade positions, θ while holding λ fixed. Mathematically this can be expressed in equation (3.29).

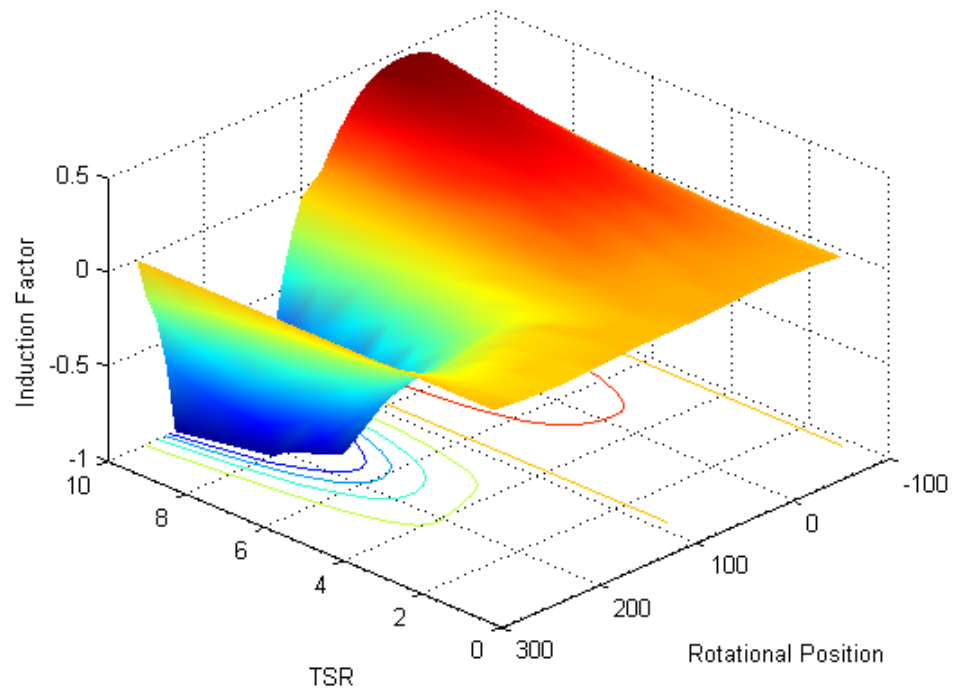
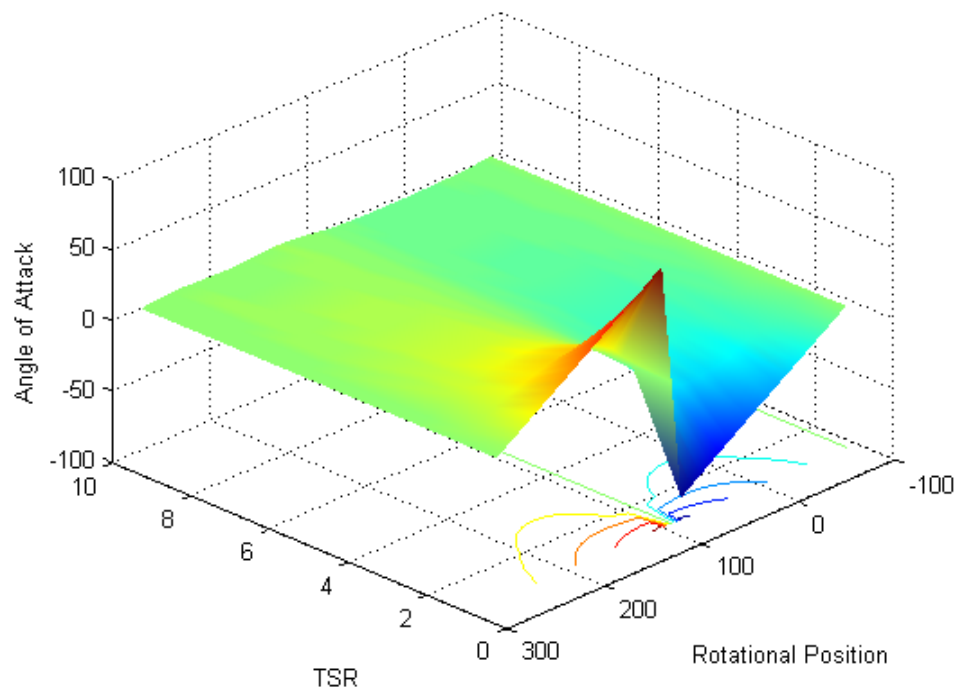
$$C_p = \frac{1}{2\pi} \int_{-\frac{\pi}{2}}^{\frac{3\pi}{2}} \frac{1}{2} p^* \cdot d\theta \quad (3.29)$$

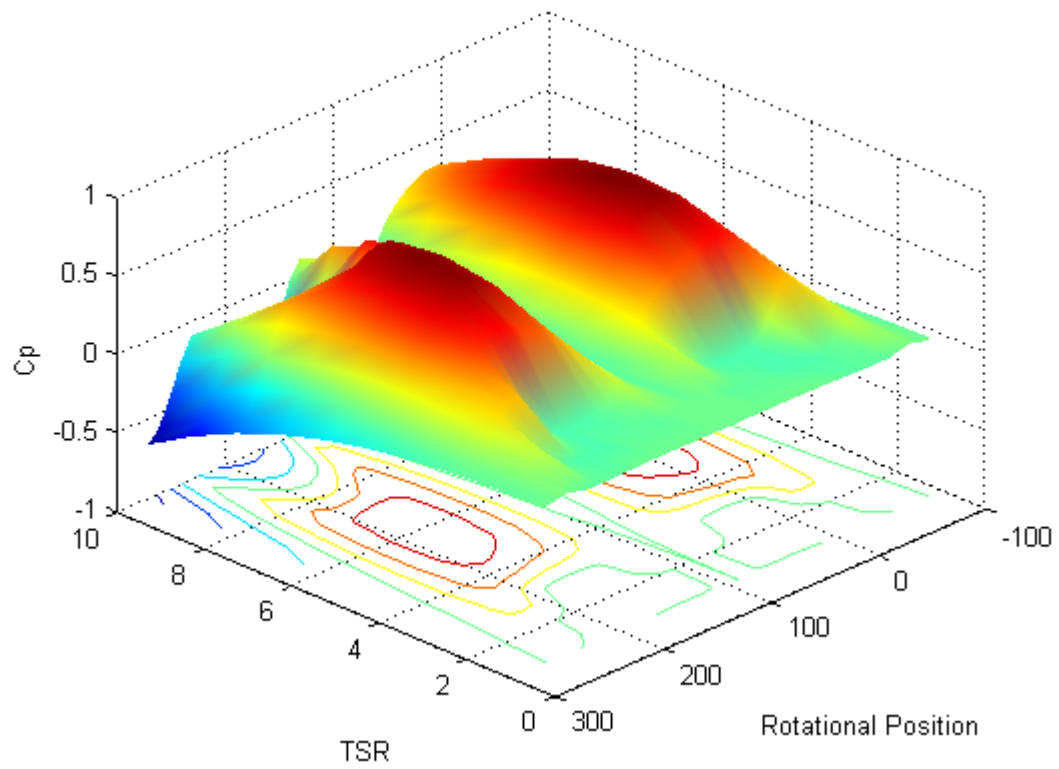
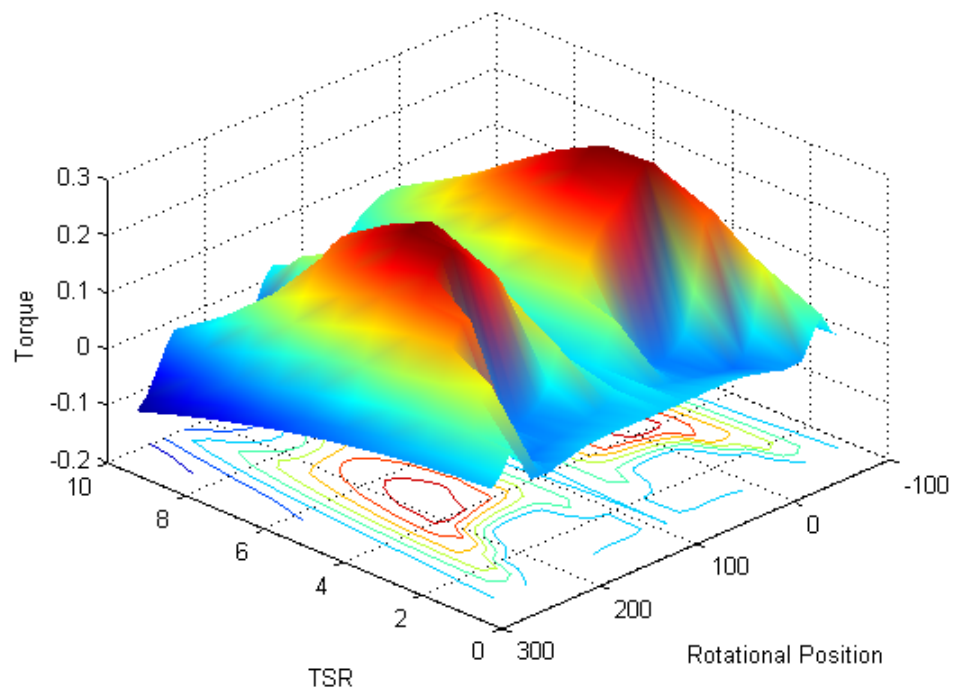
3.2.5 Circulation Control applied to a VAWT in the Momentum Model

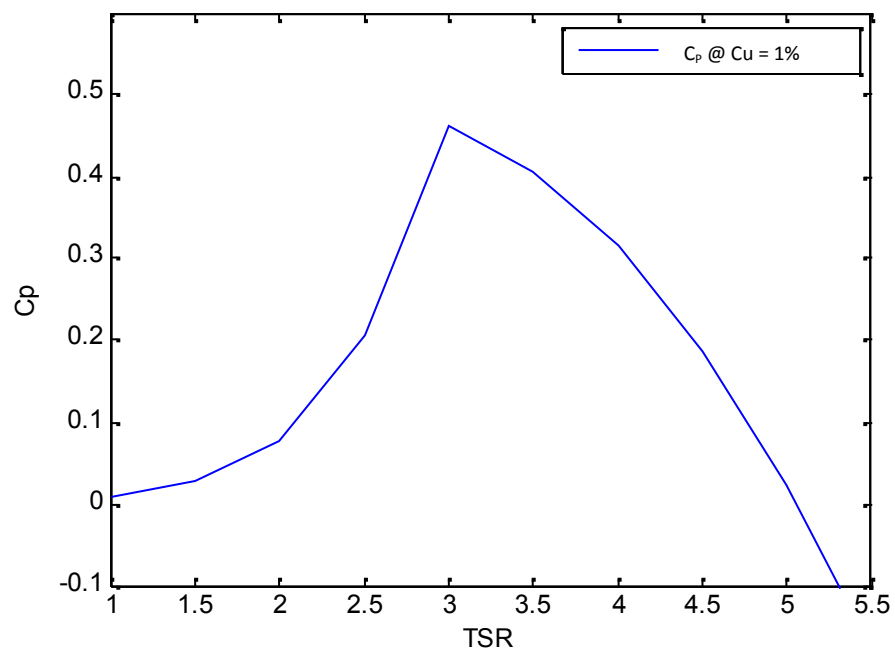
The addition of circulation control in the form of blowing slots on the upper and lower trailing edges was included in the model by using adjusted coefficients of lift and drag. The modified coefficients reflect the aerodynamic changes created by circulation control in this situation. Circulation control does add momentum and has the potential to violate the momentum balance used in this model, but for this case the added momentum from blowing was considered negligible. The coefficients of lift and drag for a circulation controlled airfoils could be similar to a NACA0018 (Sheldahl 1981), with a boost in lift and a minimal increase in drag. These coefficients for circulation control were estimated, for a preliminary study, based upon the performance of a NACA0018 (at a Reynolds number equal to 360,000) and known circulation control effects (Fanucci 1976, Angle 2008, Abramson 1975, Abramson 2004, Englar 1975). The lift coefficients are shown in Figure 3-4 and the drag coefficients in Figure 3-5, where the coefficients of lift and drag before -15 degrees and past 15 degrees are estimated to be the same as the NACA0018; these angles of attack are the approximate locations of stall for the given airfoil.

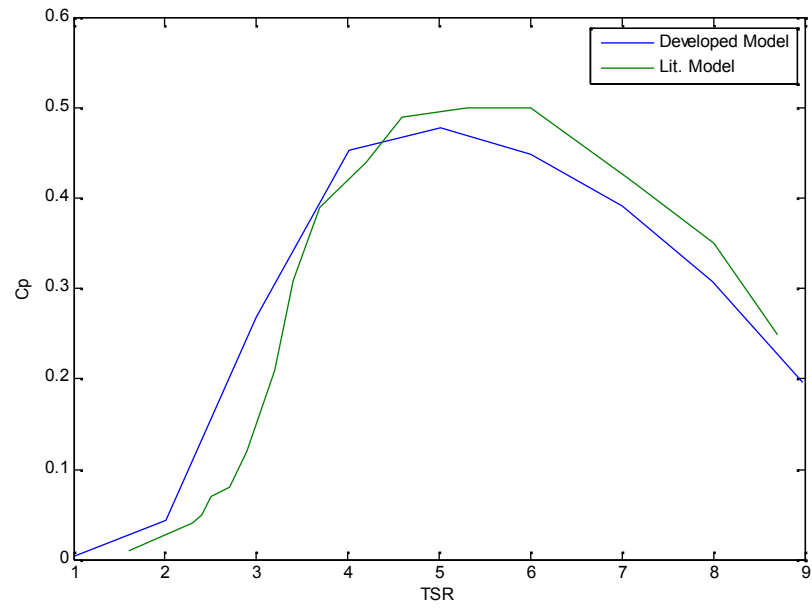


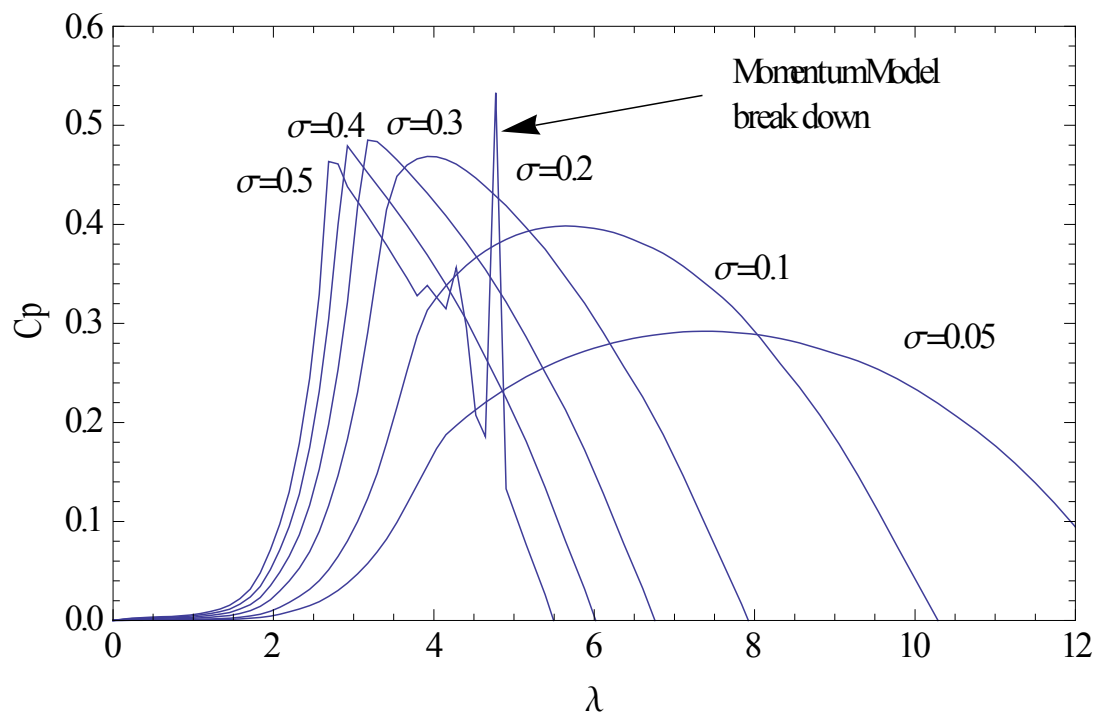












3.3 Two Dimensional Vortex Modeling

A commonly used analytical method to model the aerodynamics of a VAWT is a vortex model. Vortex models for aerodynamics are potential flow models that calculate velocities based upon bound and free wake vortices created in the wake of an airfoil. Vortex models follow the conservation of circulation by modeling discrete shed vortices whose strength is equal to the change in bound circulation over time. The vortex model was first applied to turbines by (Larsen 1975) for a windmill. Strickland and others then applied this methodology to a VAWT in two and three dimensions (Strickland 1980, Pawsey 2002, Cardone, 1984, Migliore 1978). The model discussed in this paper is based upon this prior work and was modified to include the effects of circulation control.

3.3.1 Model Implementation

The orientation of the model used to discuss its implementation can be seen in Figure 3-14. The force equation requires knowledge of the blade area, A_{bl} , the angle of attack, α , and the flow velocity, v_{bl} , as seen by the blade. The blade area is given by the chord, c , multiplied by blade length, h . It is, however, more convenient to work with wind turbine related unit-less, non-dimensional parameters such as the tip speed ratio (TSR), equation (3.30), and the solidity factor, equation (3.31), where ω is the angular velocity of the turbine with \sim denoting the variable is unit-less, r is the turbine radius and N is the number of blades.

$$\lambda = \frac{\omega r}{v_{\infty}} \quad (3.30)$$

$$\sigma = \frac{N_b C}{R} \quad (3.31)$$

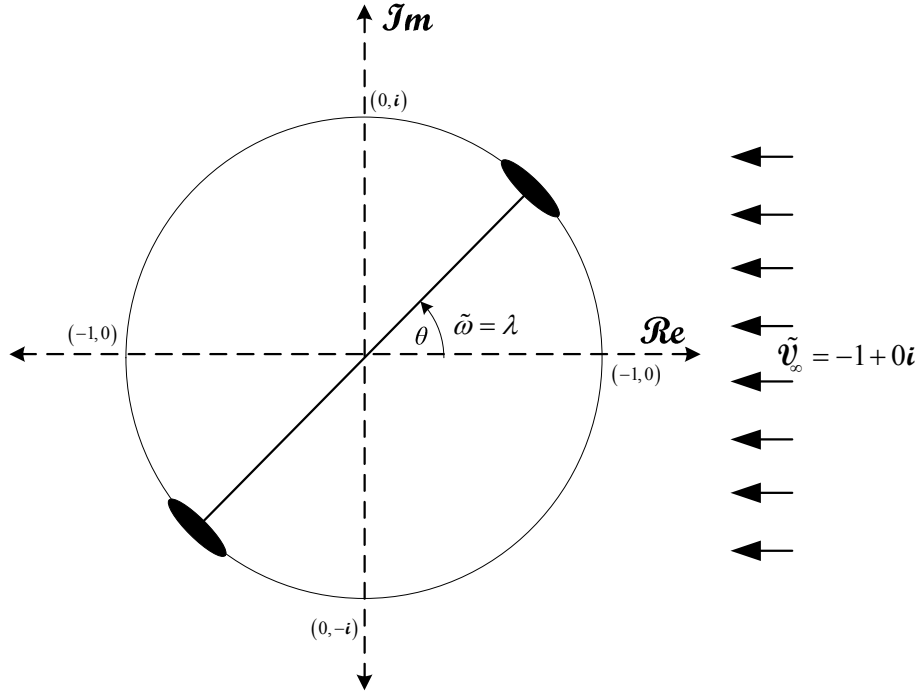


Figure 3-14 Non-dimensional VAWT Geometric Setup in the Complex Plane

3.3.2 Modeling of Airfoil Lift and Drag as complex values

The strength of the shed vortices is determined from the change in the bound blade vorticity as given by equation (3.37), where the lift and drag coefficients at the various conditions of the airfoil are needed. In addition, the normal and tangential forces, f_n and f_t , experienced by the turbine blade can be calculated from these coefficients. If the lift and drag coefficients are combined into a complex number, \mathbf{c}_{lb} , such that real part of each complex number represents the lift coefficient and the imaginary part, the drag coefficient, lift and drag can be represented by equation (3.32); \mathbf{v}_b is the relative velocity of the air with respect to the blade, script symbols denote 2D vectors (i.e. complex numbers), and * denotes the complex conjugate operation. Similarly, the normal and tangential forces are given by equation (3.33). The complex normal tangential coefficient, \mathbf{c}_{nt} , is calculated from the angle-of-attack, α , by equation (3.34).

$$(f_l + i \cdot f_d) = \mathbf{f}_{ld} = \frac{1}{2} \rho c \mathbf{c}_{ld} \mathbf{v}_b \mathbf{v}_b^* \quad (3.32)$$

$$(f_n + i \cdot f_t) = f_{nt} = \frac{1}{2} \rho c \mathbf{e}_{nt} \mathbf{v} \mathbf{v}^* \quad (3.33)$$

$$\mathbf{e}_{nt} = \mathbf{e}_{ld} e^{i\alpha} \quad (3.34)$$

3.3.3 Circulation and lift

Incompressible flow around subsonic airfoils at a density, ρ , can be modeled by the superposition of the free stream flow around the blade, \mathbf{v}_r , and a bound vortex, Γ_b , which represents the airfoil and is related to its lift, L , by equation(3.35).

$$L = \rho \Gamma_b |\mathbf{v}_r| \quad (3.35)$$

Lift of an airfoil can also be expressed in terms of the lift coefficient, C_l , and the chord length, c , by equation (3.36).

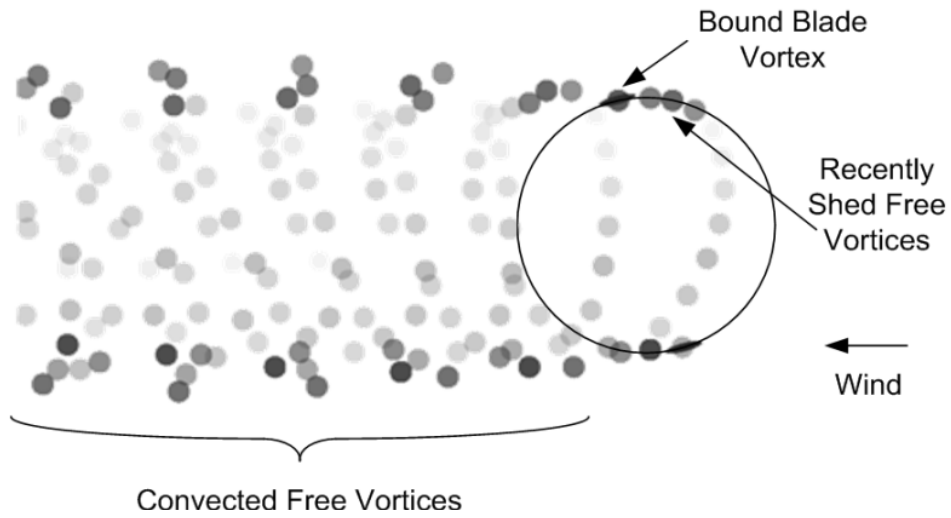
$$L = \frac{1}{2} \rho c C_l |\mathbf{v}_r|^2 = \frac{1}{2} \rho c C_l \mathbf{v} \mathbf{v}^* \quad (3.36)$$

The lift generated by an airfoil can simply be modeled as a vortex bound to the airfoil of strength Γ_b given by equation(3.37).

$$\Gamma_b = \frac{1}{2} c C_l |\mathbf{v}_r| \quad (3.37)$$

Since the lift of the airfoil changes over time due to variance in the angle of attack as the blades of a turbine rotate, the circulation bound to the airfoil will also be forced to change. This change is accommodated by shedding the difference in bound vorticity of the airfoil as a free point vortex, or a sequence of vortices for the discrete case. The free vortices are then convected down-wind of the turbine according to the local velocities at each of the vortices centers, shown by Figure 3-15. The actual flow conditions will be the sum of the free stream flow field and the induced velocities of all shed and bound vortices. The strengths of the shed vortices can be found from equation (3.38).

$$\Gamma_s = \Gamma_{b,new} - \Gamma_{b,old} \quad (3.38)$$



Δp

p_f

p_v

$$\Delta p = p_f - p_v$$

p_f

$$v_{ind} = \frac{\Delta p i \Gamma_v}{2\pi \Delta p \Delta p^*}$$

$|\Delta p|$

For a wind turbine with radius, r , spinning at angular velocity, $d\theta$, the vortex separation, ds , can be estimated by $r d\theta$, or $r\Delta\theta$ for a discrete model, resulting in a maximum core velocity of equation (3.40).

$$\max(V_{ind}) = \frac{1}{2} \frac{\Gamma_v}{r\Delta\theta} \quad (3.40)$$

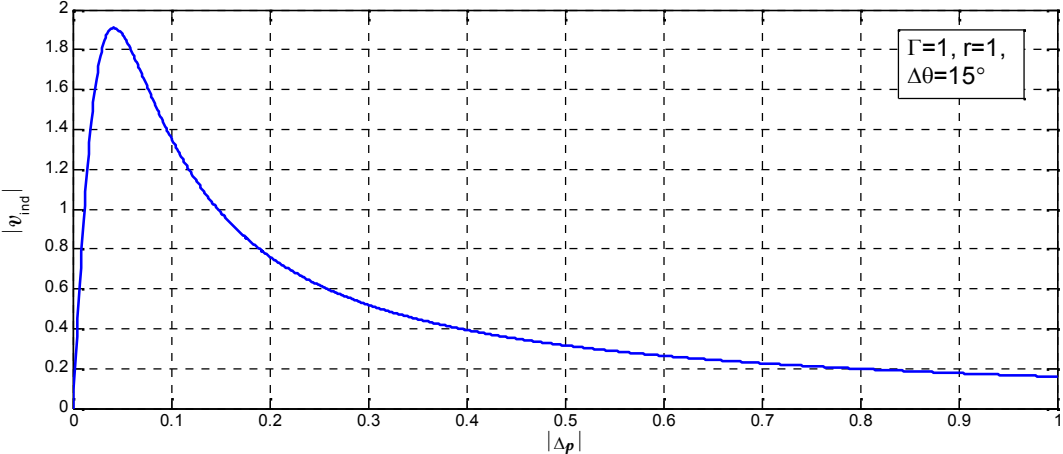
The corresponding vortex core radius, r_c , can then be determined from equation (3.39) and equation (3.40) as equation (3.41).

$$r_c = \frac{r\Delta\theta}{\pi} \quad (3.41)$$

For computational purposes, a function for vortex induced velocity (such as used by (Pawsey 2002)) is preferred to the piecewise function used by (Strickland 1980). This is achieved by modifying equation (3.39) in the denominator resulting in equation (3.42).

$$\tilde{v}_{ind} = \frac{\Delta\tilde{\rho}i\tilde{\Gamma}_v}{2\pi\Delta\tilde{\rho}\Delta\tilde{\rho}^* + \frac{(\Delta\theta)^2}{2\pi}} \quad (3.42)$$

The magnitude of the induced velocity given by equation (3.42) is plotted against distance from the vortex core with a $\Delta\theta = 15^\circ$ for illustration in Figure 3-16. The peak induced non-dimensional velocity of 1.9 occurs at a non-dimensional distance of 0.083 from the vortex center and the induced velocity falls off to zero towards the center of the vortex core.



$$\boldsymbol{v}_{flow} = \boldsymbol{v}_{\infty} + \sum \boldsymbol{v}_{ind}$$

$$\tilde{V}_{\infty}$$

$$\tilde{\boldsymbol{v}}_{flow} = (-1+0i) + \sum_k \frac{\Delta \tilde{\boldsymbol{p}}_k i \tilde{\Gamma}_k}{2\pi \Delta \tilde{\boldsymbol{p}}_k \Delta \tilde{\boldsymbol{p}}_k^* + \frac{(\Delta \theta)^2}{2\pi}}$$

$$\tilde{V}_{b_k}$$

$$\tilde{\mathbf{p}}_{b_k}$$

$$\tilde{\mathbf{v}}_{b_k} = \tilde{\mathbf{v}}_{flow_k} - i \cdot \lambda \cdot \tilde{\mathbf{p}}_{b_k} \quad (3.45)$$

$$\alpha_k = \arg \left(i \tilde{\mathbf{p}}_{b_k}^* \tilde{\mathbf{v}}_{b_k} \right) \quad (3.46)$$

Once the new relative blade velocities and angles of attack are known and recalling equation (3.37), the bound vorticity can be calculated for the k^{th} blade with equation (3.47), where \tilde{c}_k is the non-dimensional chord or chord to radius ratio of that blade.

$$\tilde{\Gamma}_{b_k} = -\frac{1}{2} \tilde{c}_k \operatorname{Re} \left\{ \tilde{\mathbf{c}}_{ld_k} \right\} \left| \tilde{\mathbf{v}}_{b_k} \right| \quad (3.47)$$

Similarly, from (Pawsey 2002), the non-dimensional normal and tangential blade forces can also be determined by equation (3.48). Once the forces are known, the blade's instantaneous coefficient of performance is given by equation (3.49). Finally, convection of the existing free vortices array occurs and the newly shed vortices are added to the previously shed vortices. Convection is implemented by equation (3.50) or in the non-dimensional case as equation (3.51).

$$\tilde{\mathbf{f}}_{nt_k} = \tilde{\mathbf{c}}_{nt_k} \cdot \operatorname{Re} \left\{ \tilde{\mathbf{v}}_{b_k} \tilde{\mathbf{v}}_{b_k}^* \right\} \quad (3.48)$$

$$C_{p_k} = \frac{1}{2} \tilde{c}_k \lambda \operatorname{Im} \left\{ \tilde{\mathbf{f}}_{nt_k} \right\} \quad (3.49)$$

$$\mathbf{p}_{v_k}(t + \Delta t) = \mathbf{p}_{v_k}(t) + \mathbf{v}_{flow_k}(t) \Delta t \quad (3.50)$$

$$\tilde{\mathbf{p}}_{v_k} \left(\tilde{t} + \frac{\Delta \theta}{\lambda} \right) = \tilde{\mathbf{p}}_{v_k}(\tilde{t}) + \tilde{\mathbf{v}}_{flow_k}(\tilde{t}) \frac{\Delta \theta}{\lambda} \quad (3.51)$$

This vortex method, when implemented using the previous equations and methods, will provide an analytical model which will be used to predict the blade forces and VAWT performance. Also, the

model allows parameters such as the blade lift and drag, turbine solidity, radius, and TSR to be selected for their respective ranges of operation. The blade force and turbine performance for a wide range of TSR, solidities, and different blades will then be used to develop a control scheme for maximum power capture using circulation control.

3.3.5 Model Convergence

Evaluating the vortex model begins by initializing the velocity field to free-stream conditions, \tilde{V}_∞ , meaning that no vortices have been created. The convergence process can take a few iterations to occur. Convergence may also never occur as in oscillatory or chaotic cases. A check for convergence was performed after four rotations of the turbine. The maximum number of rotations was limited to 60. The convergence criteria was a comparison of the percent difference of the last revolution's average C_p to the previous revolutions C_p to a threshold of 0.1%.

The turbine motion was divided into $\Delta\theta$ steps (typically 24 as suggested by (Strickland 1979)) which determine the number of vortices which are shed during rotation. One full rotation calculation occurs until the coefficient of performance can be evaluated. The higher the number of $\Delta\theta$ steps the greater computation time. In addition to the increase of computation time for more $\Delta\theta$ steps, more time is spent determining the blade velocity since there are more vortices in the range of the turbine with a higher $\Delta\theta$ step.

3.3.6 Dynamic Stall in the Vortex Model

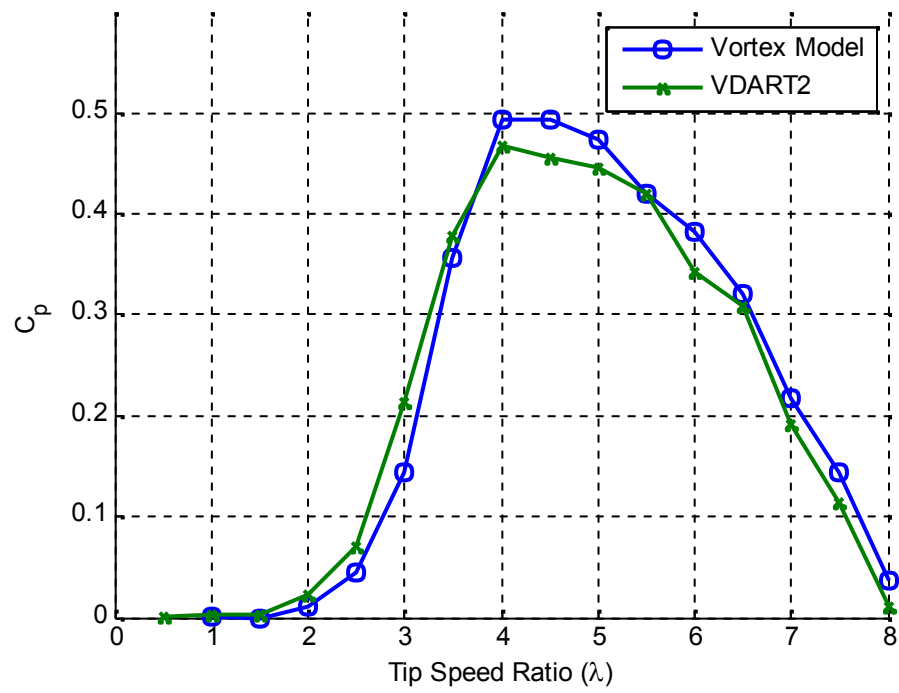
The rotational motion of a VAWT causes the blade to continually experience a variation of angle of attack (α) with the angle of rotation (θ); The rapid positive and negative deviation of the blade's angle of attack, otherwise known as pitching motion, can invoke unsteady dynamic behavior in the flow field. The phenomenon of dynamic stall is thus an inherent potential effect for the recurrent blade pitch (Leishman 1990, Hansen 2001). Dynamic stall refers to the stalling of an airfoil when the angle of attack

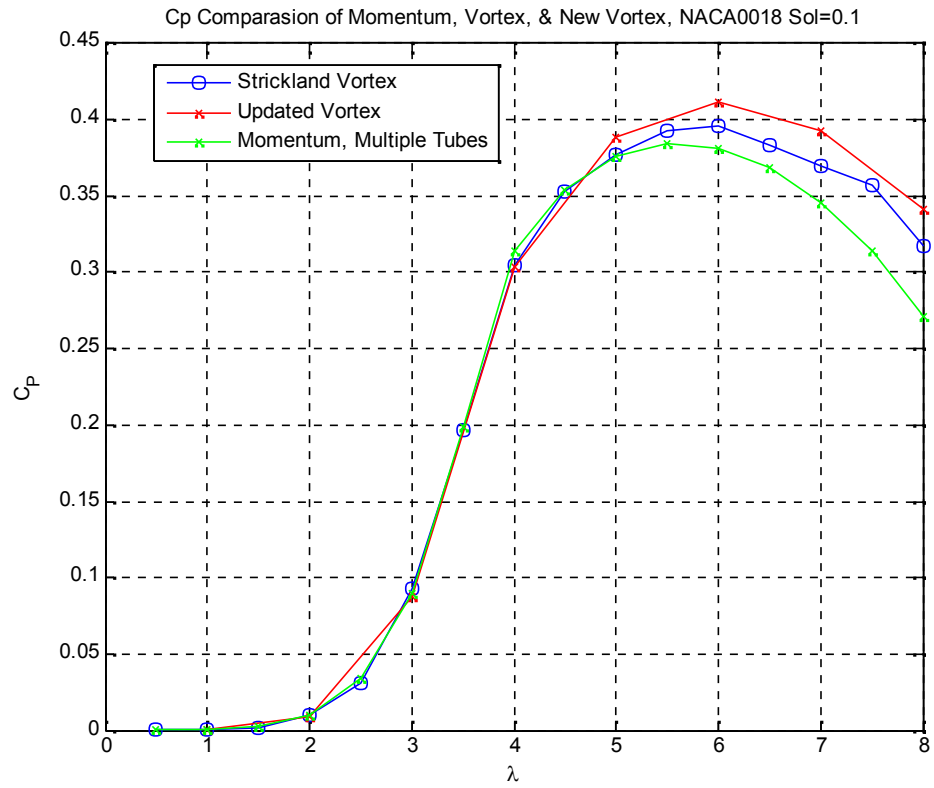
is changing rapidly with time. At low λ , the blade experiences the largest range of angle of attack and dynamic stall is thus more prevalent. The pitch related change has shown to have small impacts on turbine performance (Paraschivoiu 1988).

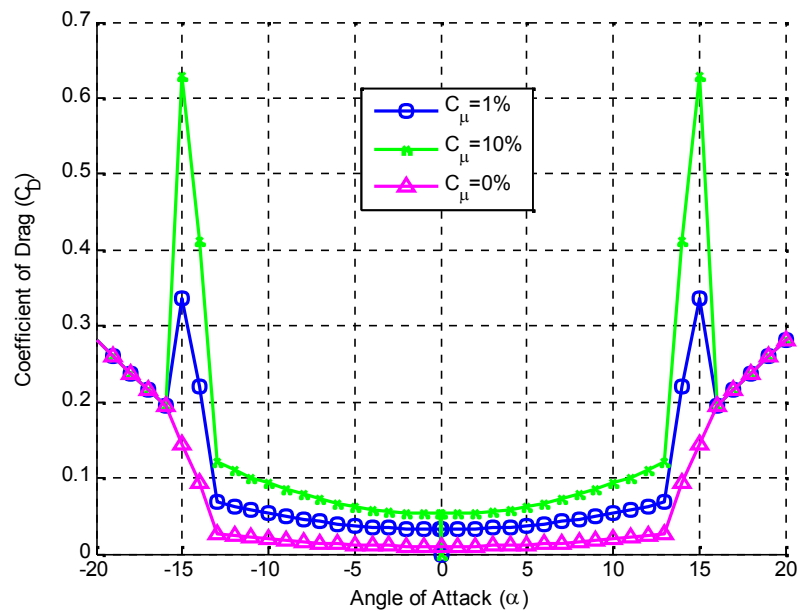
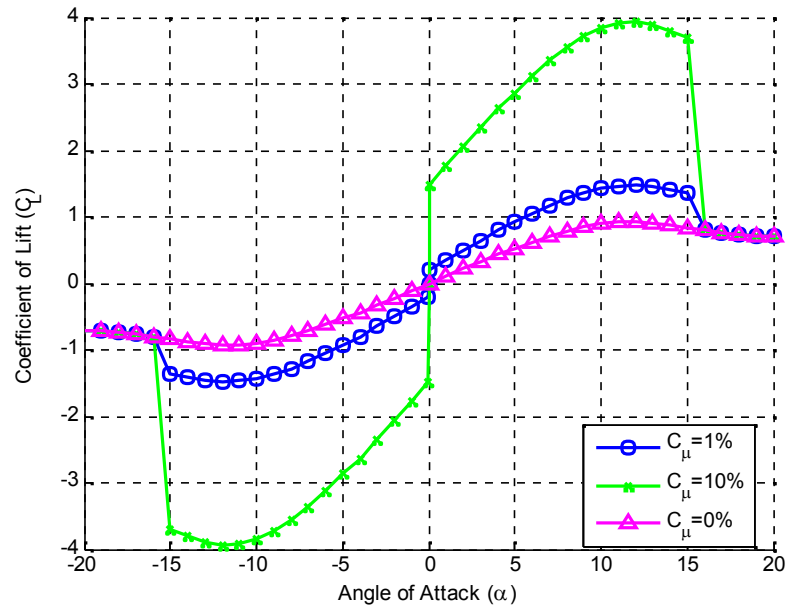
The vortex model implementation described here does not currently incorporate dynamic stall. The effects of circulation control and dynamic stall are currently unknown. To properly model dynamic stall, the specifics of the blade during stall are required. When a numerical study or experiment are performed on a full scale prototype to gather this information then it can be implemented into the vortex model to help refine the model outcomes.

3.3.7 Vortex Model Validation

Comparison of the model implementation developed here to a previously created vortex model computer code was performed by matching parameters. Strickland created a 2D vortex model with a solidity of 0.2 and two NACA0012 (Sheldahl 1981) blades. The resultant coefficient of performance is shown by Figure 3-17, where “VDART2” can be found in (Strickland 1980) and the “Vortex Model” as described here. Despite using potentially differing airfoil data for the NACA0012, the trends are similar enough to validate the general behavior of VAWT behavior and provide to insight.







θ

λ

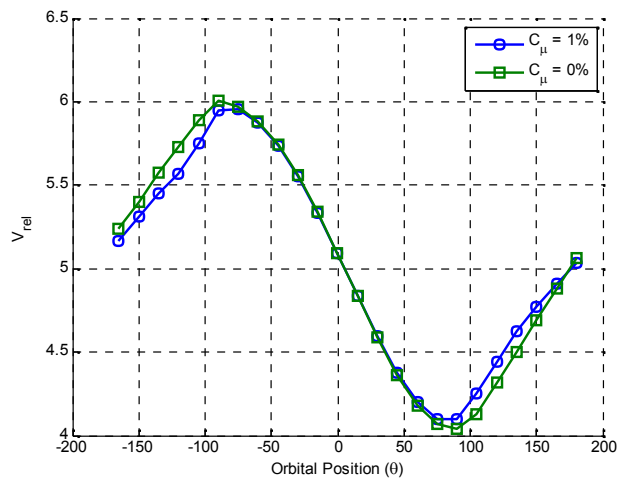


Figure 3-21 Vortex Model Relative Velocity TSR=4

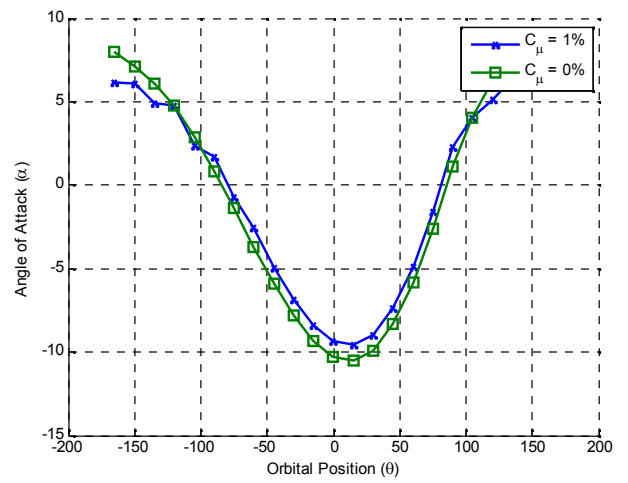


Figure 3-22 Vortex Model Angle of Attack TSR=4

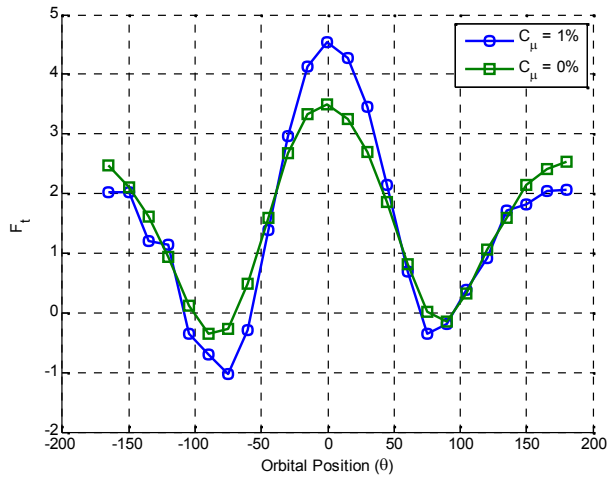


Figure 3-23 Vortex Model Tangential Force TSR=4

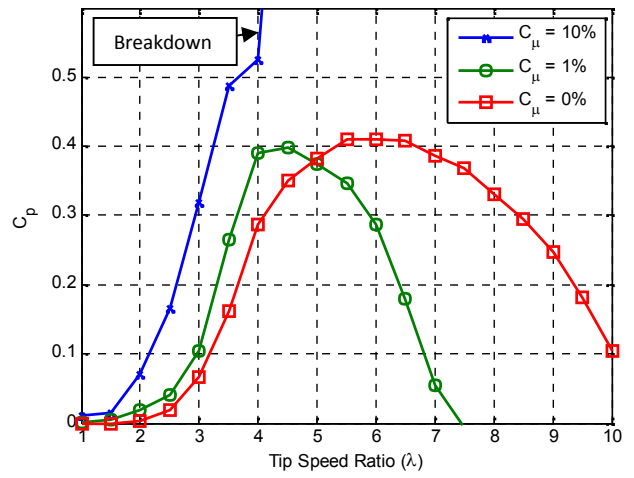
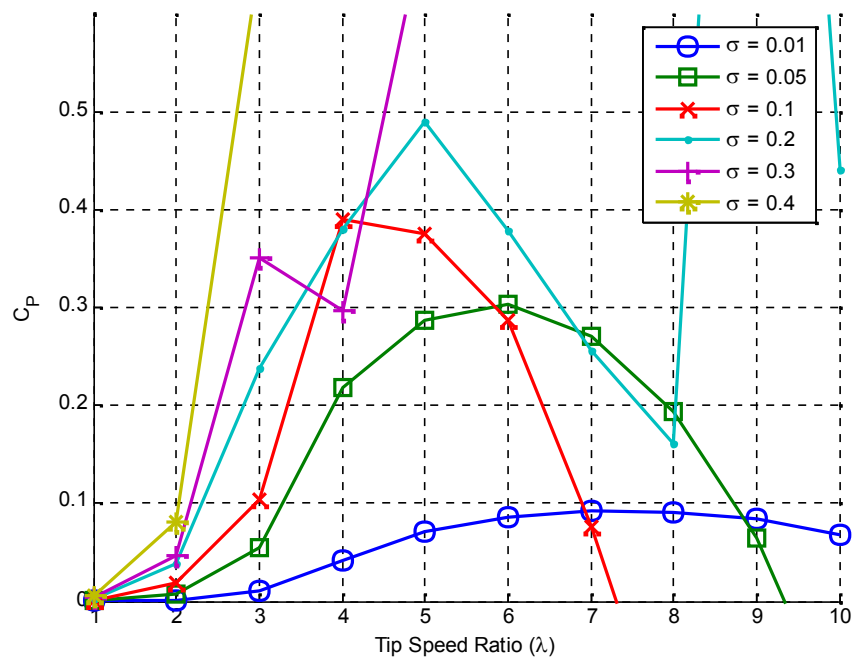
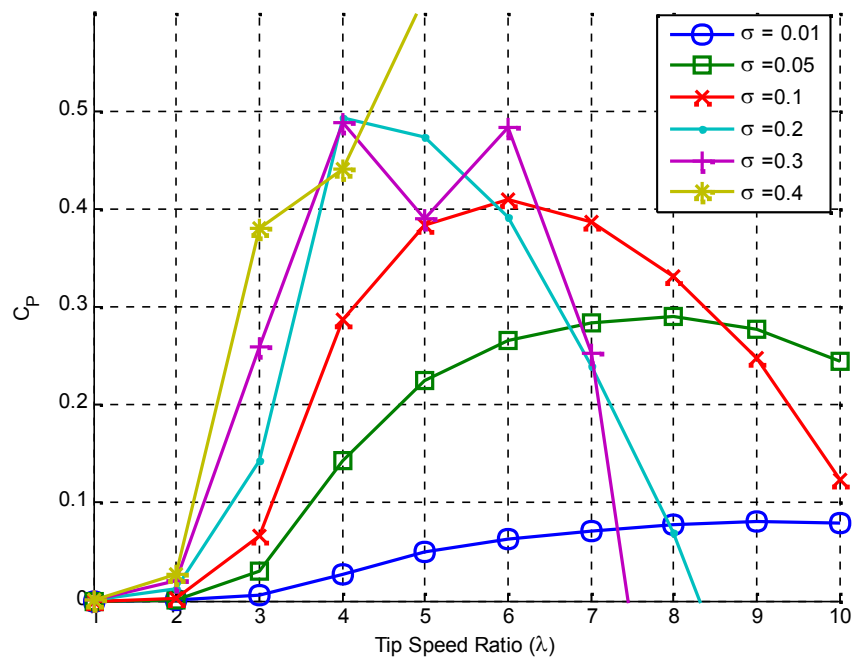
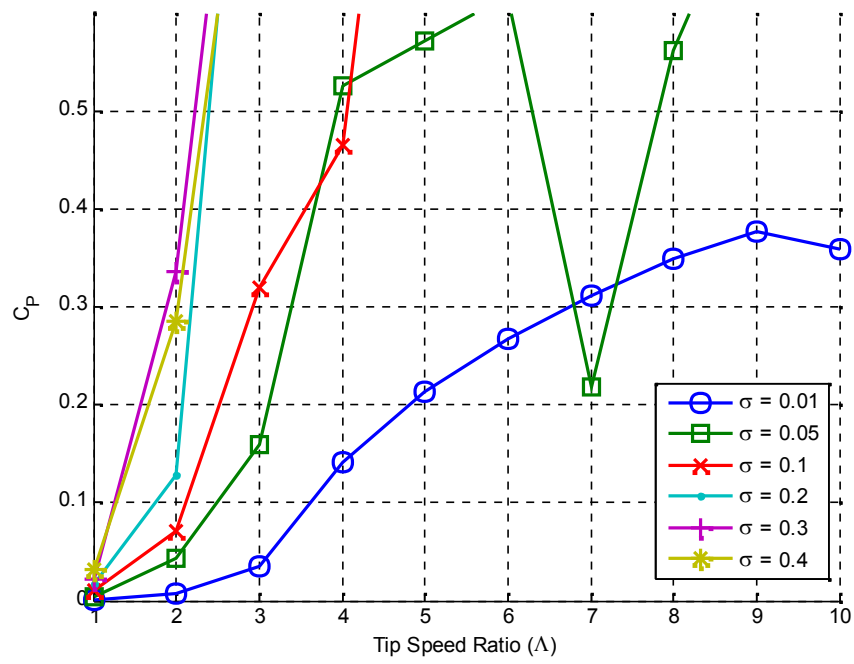
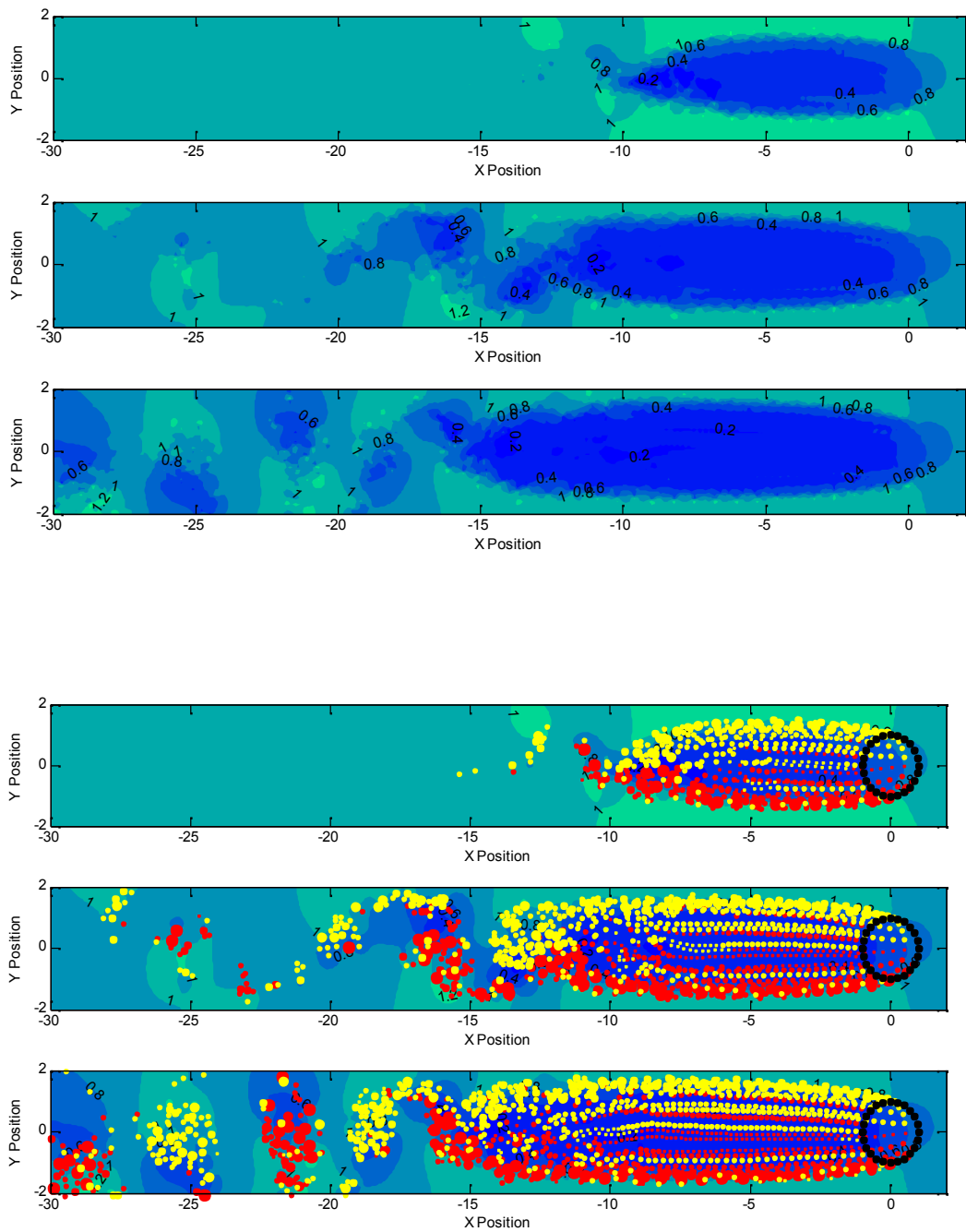
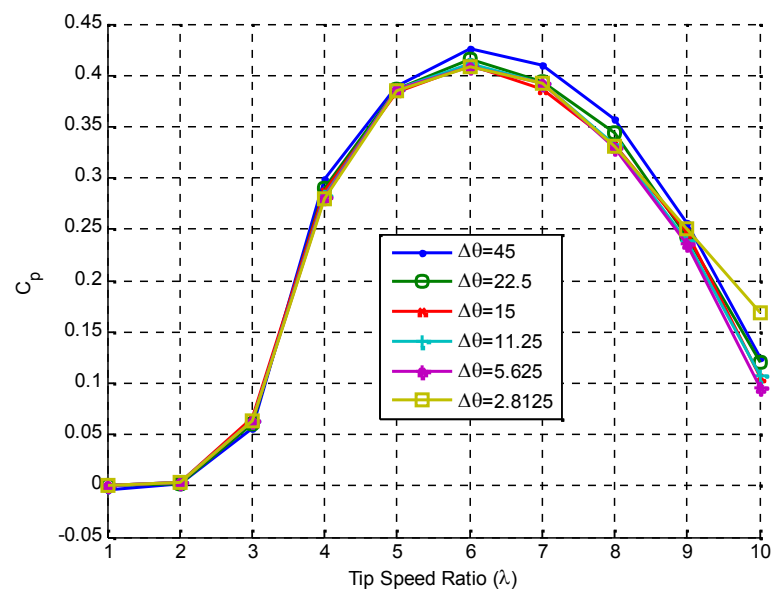


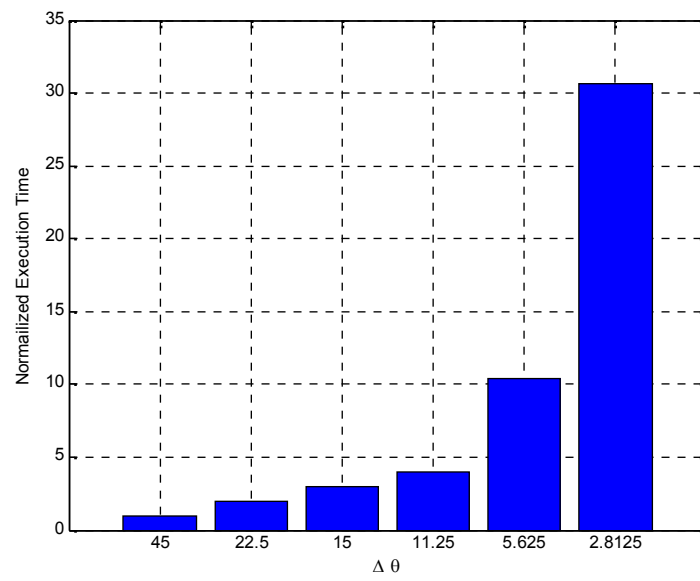
Figure 3-24 Coefficient of Performance

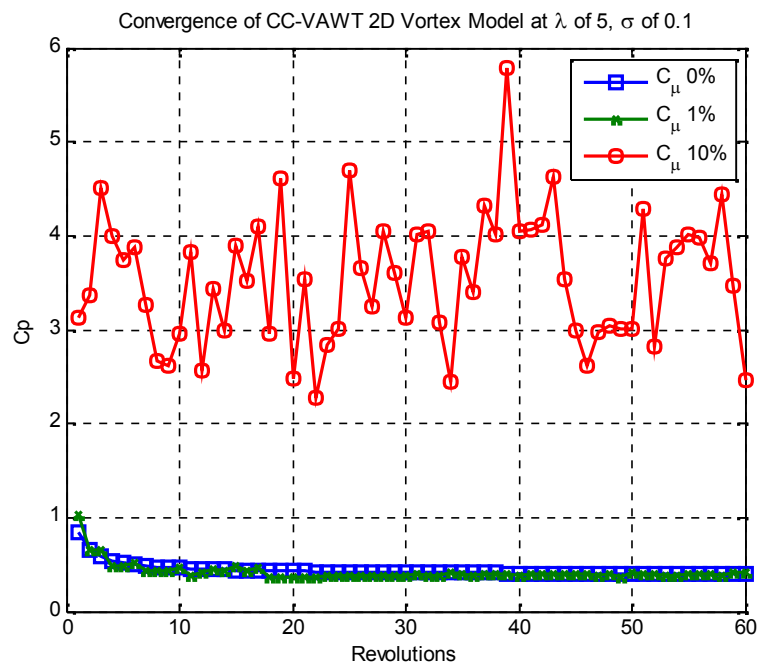
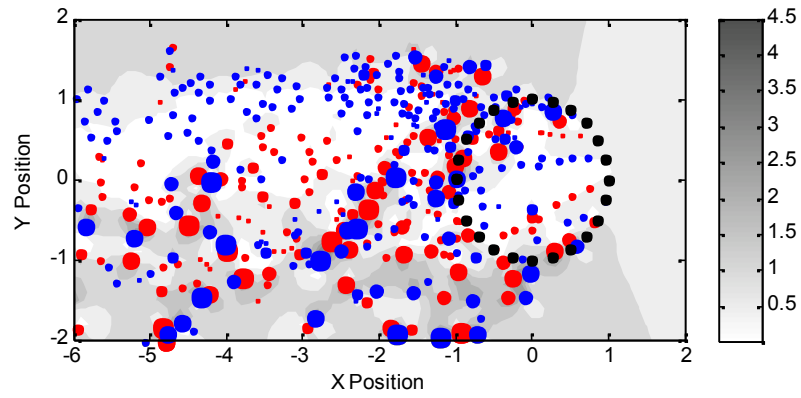












testing of a scaled and full size prototype design based on these initial results will be completed in the future to further refine the current analytical model.

A 2D model of a CC-VAWT was developed and evaluated using preliminary circulation control airfoil data for a CC-VAWT blade. Further examination and experimentation on the effects of circulation control using the developed model will provide verification for correlation of the blowing coefficient and the apparent solidity.

3.4 Three Dimensional Vortex Modeling

A 3D vortex model is very similar to the 2D version, except the span of the blade is divided into separate elements. In the 2D model, a bound vortex is shed in the direction of the blade velocity. The 3D version also sheds this vortex, but sheds two more vortices normal to each blade element tip. 3D Vector math is used in place of a complex number scheme, used in the 2D model, to represent the vortices points and circulation. Because three vortices are shed at each iteration per blade segment, instead of a single vortex for the 2D case, more computations are required to achieve convergence. The 3D vortex model also takes into account tip losses.

To compute the 3D velocity induced by another point vortex, the following bound vortex equation is used

$$V_p = \Gamma \frac{\vec{c} \otimes \vec{a}}{(4\pi(\vec{c} \otimes \vec{a}) \bullet (\vec{c} \otimes \vec{a}) + \delta)} \left(\frac{\vec{a} \bullet \vec{b}}{|\vec{b}|} - \frac{\vec{a} \bullet \vec{c}}{|\vec{c}|} \right), \quad (3.52)$$

where Γ is the circulation for the vortex, δ is the angular step. A small constant value δ is used in equation (3.52) to resolve a problem of infinite induced velocity, something that does not occur in reality. Limiting the velocity of the vortex leads to a viscous core vortex model. The change in circulation along the blade trailing vortex sheet allows the determination of the core maximum velocity as a

function of vortex element spacing set by the discrete angular step increments of turbine rotation, $\Delta\theta$ and the number of blade elements. The δ value can be expressed as

$$\delta = \frac{\Delta\theta^2}{2\pi}, \quad (3.53)$$

where,

$$\Delta\theta = \frac{\pi}{2} \frac{\tilde{h}}{N_b}. \quad (3.54)$$

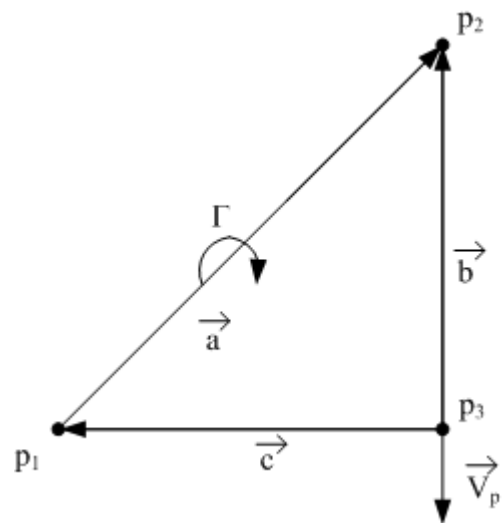
The vectors a , b , and c are represented as

$$\vec{a} = \vec{p}_2 - \vec{p}_1; \vec{b} = \vec{p}_2 - \vec{p}_3; \vec{c} = \vec{p}_1 - \vec{p}_3 \quad (3.55)$$

or

$$\begin{Bmatrix} \vec{a} \\ \vec{b} \\ \vec{c} \end{Bmatrix} = \begin{bmatrix} -1 & 1 & 0 \\ 0 & 1 & -1 \\ 1 & 0 & -1 \end{bmatrix} \begin{Bmatrix} \vec{p}_1 \\ \vec{p}_2 \\ \vec{p}_3 \end{Bmatrix}. \quad (3.56)$$

The points p_1 and p_2 represent the endpoints of the vortex and p_3 represents the point to calculate the induced flow, \vec{V}_p . A graphical representation for the location of the vortex points and the induced point is shown in Figure 3-34.



$$V_{\infty}$$

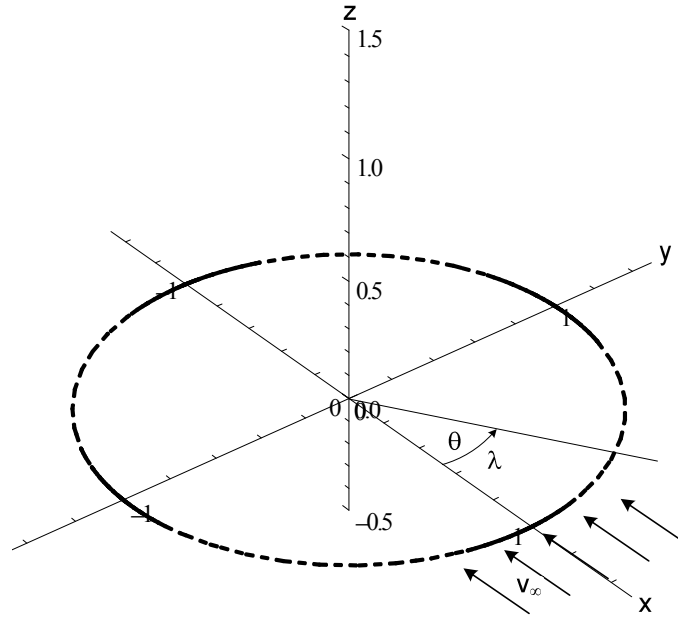


Figure 3-35 Non-dimensional 3D VAWT geometric setup

The angle-of-attack for the 3D vortex model is calculated using the relative blade element velocity, a normal vector (\vec{N}), and chord vector (\vec{C}) in the following equation

$$\alpha = \tan^{-1} \left\{ \frac{\vec{V}_{rel} \cdot \vec{N}}{-\left(\vec{V}_{rel} \cdot \vec{C}\right)} \right\} \quad (3.57)$$

Circulation of the newly formed bound vortex can be calculated by

$$\Gamma_b = \frac{1}{2} \frac{C}{R} \vec{S}_L \sqrt{\vec{V}_{rel} \cdot \vec{V}_{rel}} \quad (3.58)$$

where \vec{S} is the unit span vector. The tangential blade force is calculated by

$$F_T = \frac{1}{2} c_t \frac{C}{R} \vec{S} \sqrt{\vec{V}_{rel} \cdot \vec{V}_{rel}} \quad (3.59)$$

Finally, the coefficient of performance is calculated from the tip speed ratio, chord to radius ratio, and tangential force by

$$C_P = \lambda \frac{C}{R} F_T \quad (3.60)$$

The turbine blades are subdivided into a number of blade elements, each of which is modeled by a lift generating bound vortex line and trailing tip vortex lines at each end of the blade element. If the blade element experiences a change in angle of attack, causing lift and the bound circulation, Γ_b , of the blade element to change, the tip vortices circulation are adjusted together with the shedding of a blade element on the span-wise vortex line. The span-wise vortex line element is assigned by difference in circulation experienced by the blade element. Note that the tip vortex line elements of adjacent blade elements sum or subtract depending on their direction. This is illustrated below in Figure 3-36, where n denotes a discrete time index.

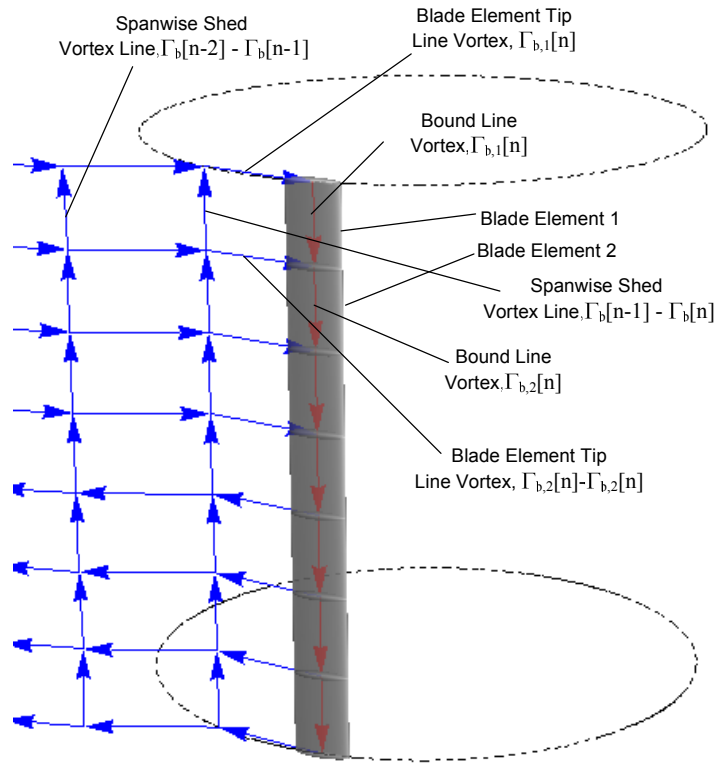


Figure 3-36 Blade subdivision into blade elements and associated vortex line elements.

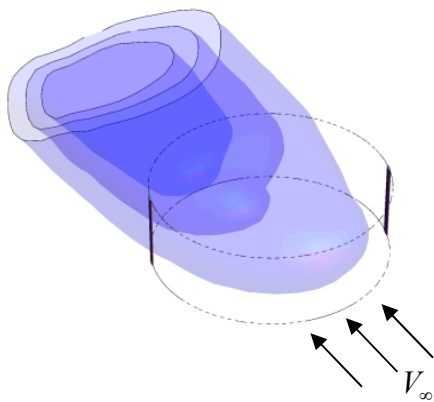


Figure 3-37 Constant velocity magnitude surfaces at 0.95, 0.9 and 0.85 V_∞ for $\lambda=4$.

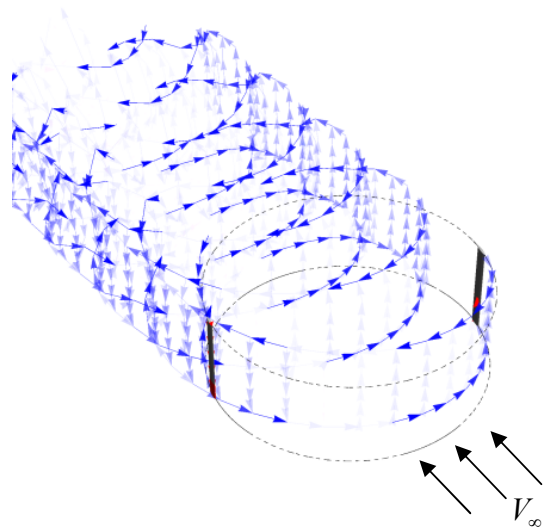
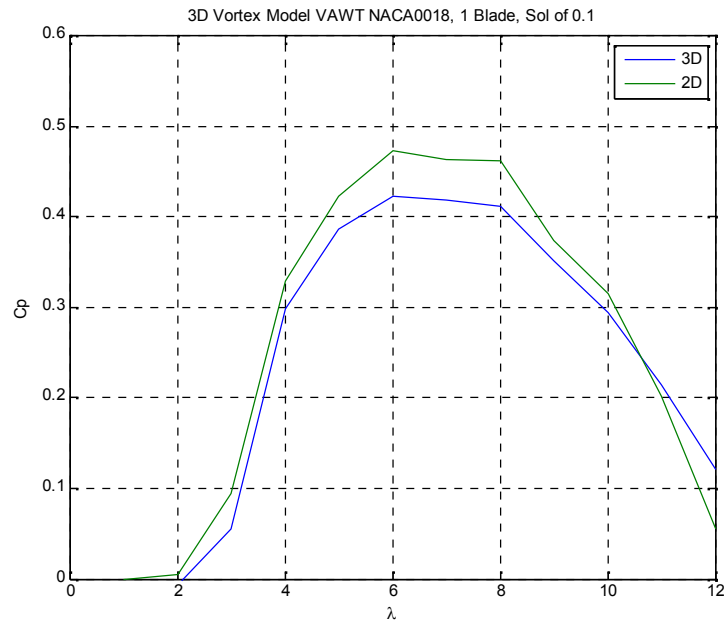


Figure 3-38 Vortex filament wake for TSR of 4 after 5 revolutions



general purpose. The two dimensional and three dimensional models subroutine which determines the point velocity was ported to run on a specific graphics card.

3.5.1 CUDA

NVIDIA produced an Application Programmable Interface (API), which allows a programmer to utilize the Graphical Processing Unit (GPU) for general purpose code execution. The NVIDIA GPU is a set of many execution units that operate on one set of code but different data. Unlike the Central Processing Unit (CPU), a GPU has minimal computational functionality and access to smaller faster memory. Because of these advantages many GPU execution units can be placed in the same physical space as a CPU, but these GPU units must be specially programmed for parallel execution. The ability of the GPU to run parallel programs for a decrease in computational time has much potential. A sampling of which can be seen at (NVIDIA CUDA 2009).

Currently, 'C' is the only language that can directly communicate with the CUDA API. This interface allows a programmer to implement any function written in 'C' to be executed on the graphics card. During compile time a parsing program first reads the 'C' source file (.cu) and translates the sections of code that will run on the GPU to a GPU compatible binary format and inserts the appropriate stubs which load this code when called. Next, CUDA passes the file onto a specified 'C' compiler which will compile the code to a CPU compatible format just like regular code would be compiled.

The two-dimensional and three-dimensional vortex models were both implemented in MATLAB, which currently has no native support for CUDA. MATLAB does have an interface to communicate with programs written in 'C', called a MEX function. The MEX function is built using any standard 'C' compiler that can develop code libraries. In this case, Microsoft Visual Studio 2008 was used to develop a 64-bit MEX DLL that MATLAB can call upon. The MEX library built for the vortex model velocity function takes the inputs and converts them from a double floating point to a single precision floating point and copies

them from the CPU to the GPU memory. The older NVIDIA hardware, prior to the 200 GTX series, can only work with single precision floating point variables, therefore double precision variables must be converted. The MEX library then calls the GPU code, which is run in parallel, and waits for the result. Once the GPU is finished executing the result is copied from GPU memory back to the CPU memory. Then the result is changed back to a double precision format, the native MATLAB format, and a pointer to that data is returned to MATLAB. An overview of the process is displayed as a flowchart in Figure 3-40.

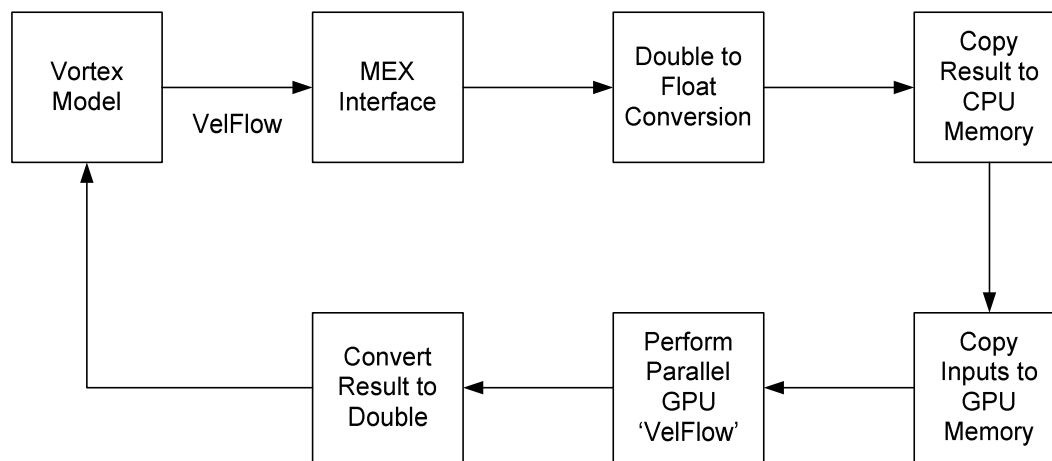


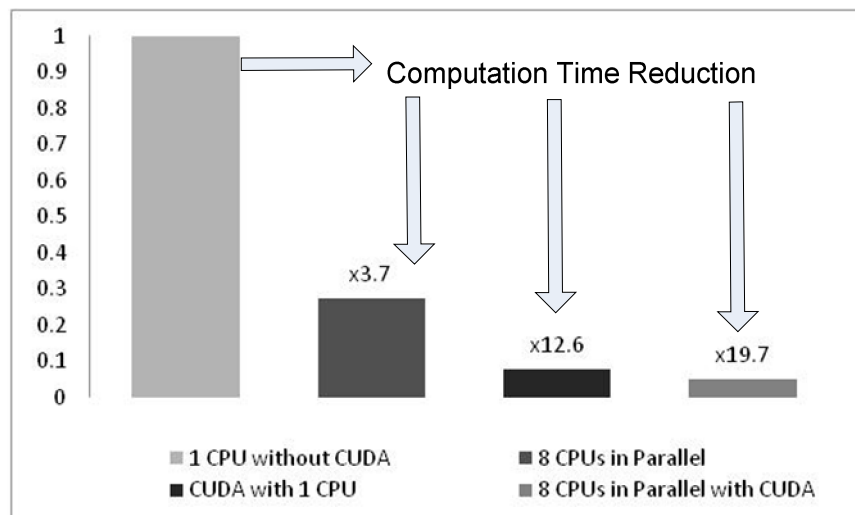
Figure 3-40 Flow Chart of CUDA Vortex Model Speedup

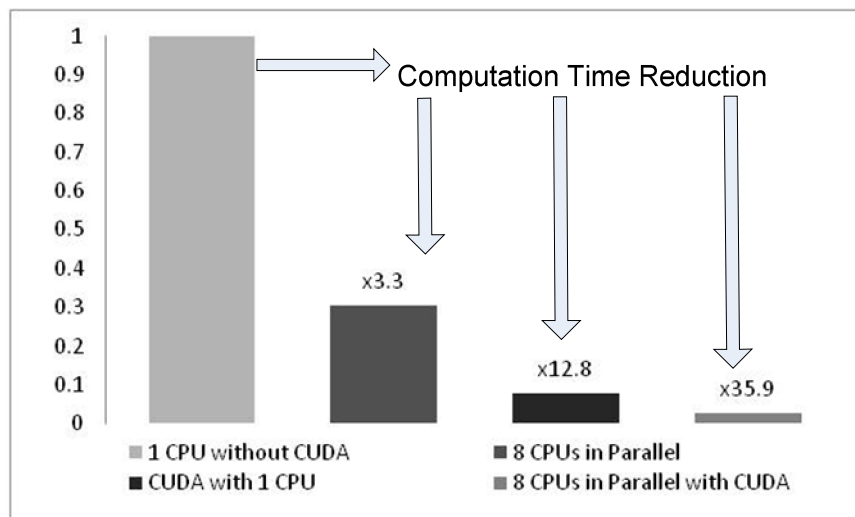
3.5.2 Computational Speed Increase Results

For this performance analysis, a computer with two Intel Xeon processor E5430, a 2.66 GHz quad core processor with a front side bus speed of 1333MHZ, was used. The operating system used was Windows Vista 64bit with 16 gigabytes of DDRAM and a NVIDIA GeForce GTX 285 graphics card. Four experiments were done with this PC setup. The tests involved benchmarking the models which included: execution on a single processor core, on all eight of the processor cores in parallel, execution on the NVIDIA graphics card, and execution on both the graphics card and both quad core processor in parallel.

Four test cases were run for the 2-D and 3-D models. Averages of the five runs were used to compare each of the four computer configurations. Table 1 contains the time required for each

Case:	2-D		3-D	
	(s)	(%)	(s)	(%)
1 CPU no CUDA	168.29	-	2919.10	-
8 CPUs no CUDA	46.06	365.40	889.04	328.34
CUDA with 1 CPU	13.31	1264.10	227.47	1283.30
CUDA with 8 CPUs	8.55	1967.75	81.33	3589.21





3.6 CC-VAWT Modeling Conclusion

Analytical modeling of a VAWT was achieved using a vortex model which predicts the blade angle of attack, velocity, torque, and coefficient of performance for a given turbine with known solidity and static aerodynamic coefficients. The effects of circulation control were added to the model to investigate its effect on performance. The developed momentum and vortex model produced results with comparable trends of past models, providing sufficient validation. In conclusion, two fundamentally different models of CC-VAWT aerodynamics were developed and evaluated using preliminary airfoil data. This model will be used to develop performance predictions and test control algorithms for a CC-VAWT provided in the following chapter.

Chapter 4. CC-VAWT Blade Characteristics

4.1 Introduction

Analytical modeling of a VAWT relies upon static blade characteristics such as lift and drag. A VAWT can experience large angle-of-attack changes during rotation, ± 180 degrees, and static data is needed for these regions. The static lift and drag coefficients and the response time of the blade internal pneumatic valves for a CC-NACA0018 were determined by an experiment that was performed in the WVU closed loop wind tunnel (Panther 2010). Pneumatic valves were used inside of the airfoil to turn on or off blowing. Load cells and pressure taps were used to measure lift and drag at a Reynolds number of 160k and 300k for a range of angles-of-attack and 0%, 1%, 5%, and 10% blowing.

Equation Chapter 4 Section 1

Two models were constructed for testing. The first model was built of metal with a fiber glass skin and its cross section was similar to that of a NACA0018, but contained a rounded trailing edge at 88% of the chord. The rounded trailing edge had a radius of 3.125% of the chord. This model was used for steady-state blowing coefficient testing only. The second model, built using a rapid prototyping machine, consisted mostly of plastic parts and contained 14 pneumatic valves. Both of the models had a 0.2 m (8 in) chord, the same as the VAWT experimental device, and were 0.41 m (16.25 in) in span and contained a single blowing slot on the upper surface.

4.2 Measurement of Lift and Drag

The lift and drag of the CC-NACA0018 blade were measured with a load cell setup, for both models, and a pressure scanning device for the rapid prototype model. The load cell system consisted of four load cells in two groups. Two cells were placed at the top and bottom of the support structure as shown in Figure 4-1. One load cell at each location was placed in the lift direction and the other in the

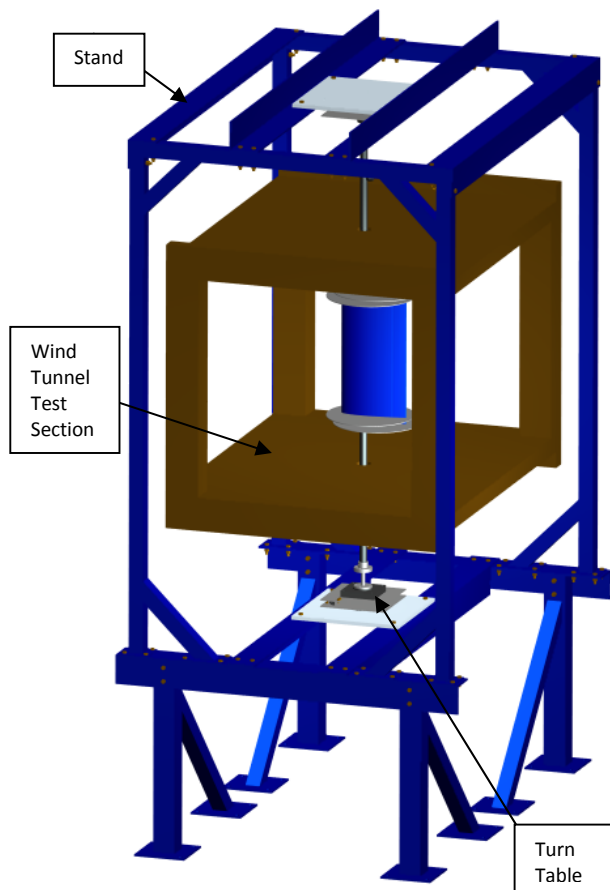


Figure 4-1 Model Stand with Model and Wind Tunnel Test Section

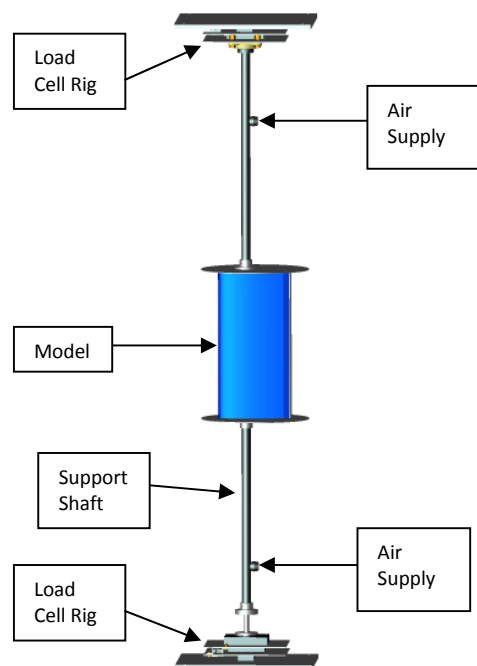


Figure 4-2 Wind Tunnel Model Support and Mounting System

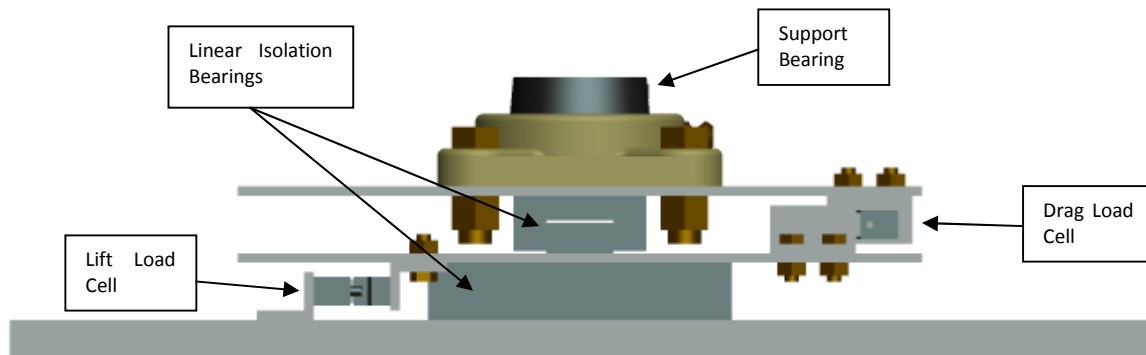




Figure 4-4 CC-NACA0018 CAD Drawing of the Metal Support Structure

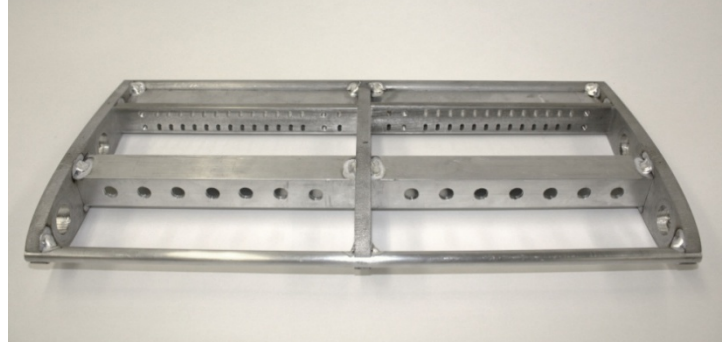


Figure 4-5 CC-NACA0018 Assembled Support Structure

The CC NACA 0018 model can be seen fully mounted and ready for testing in WVU's Subsonic Closed Loop Wind Tunnel (Figure 4-6).

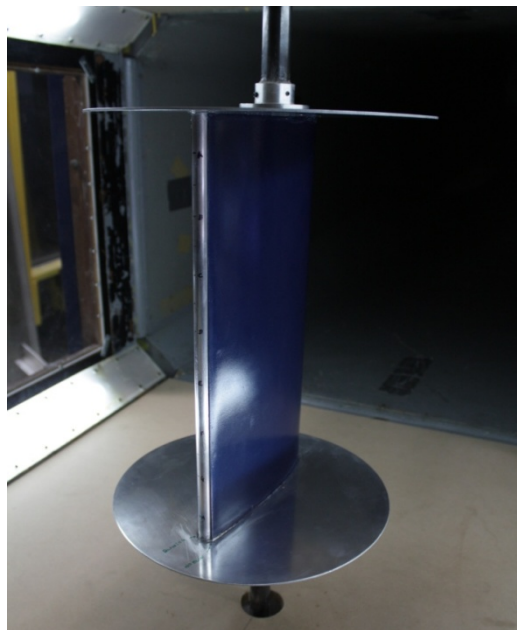
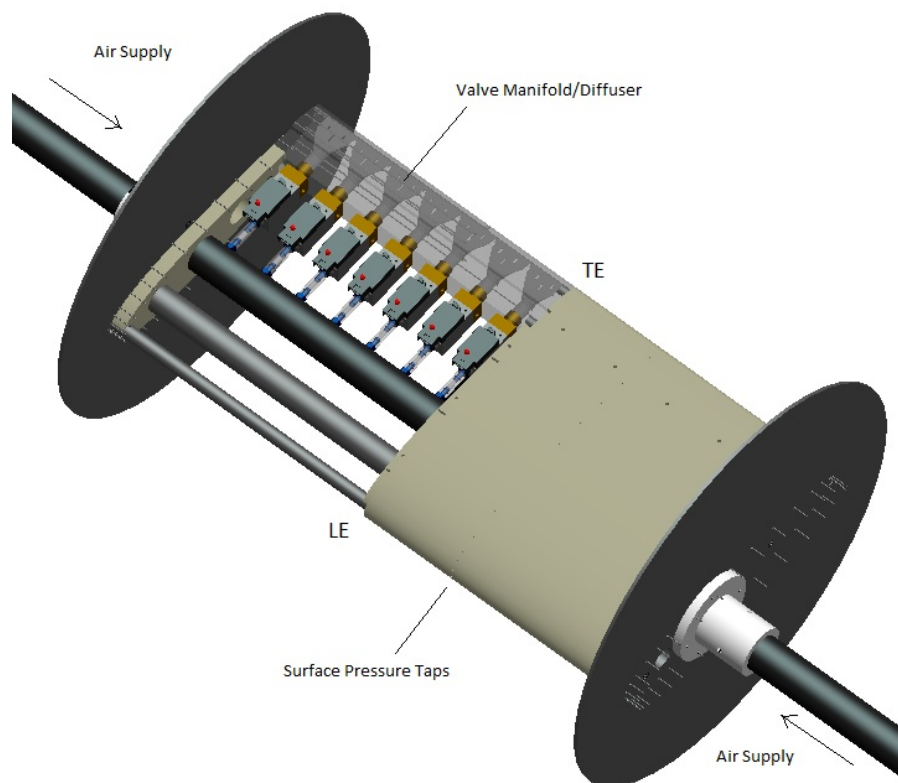
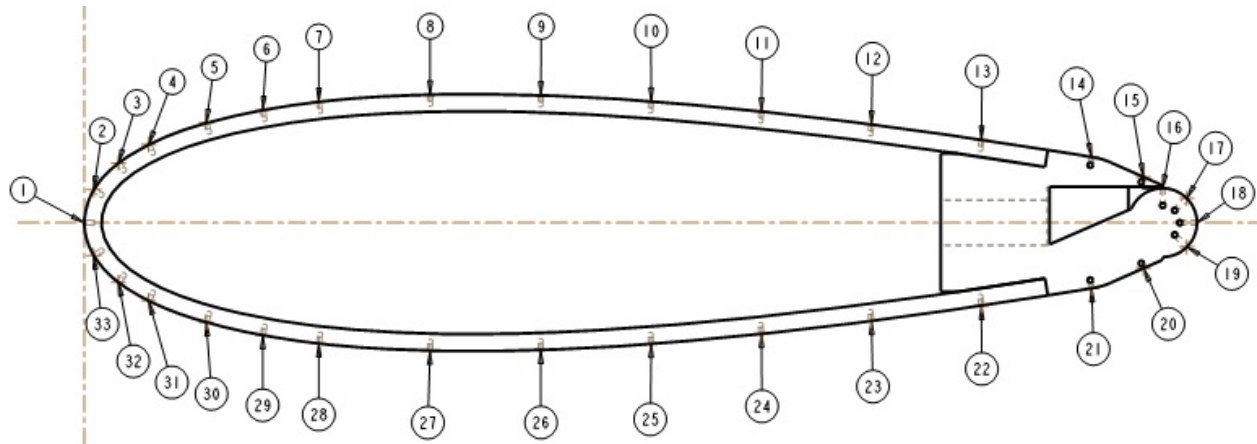


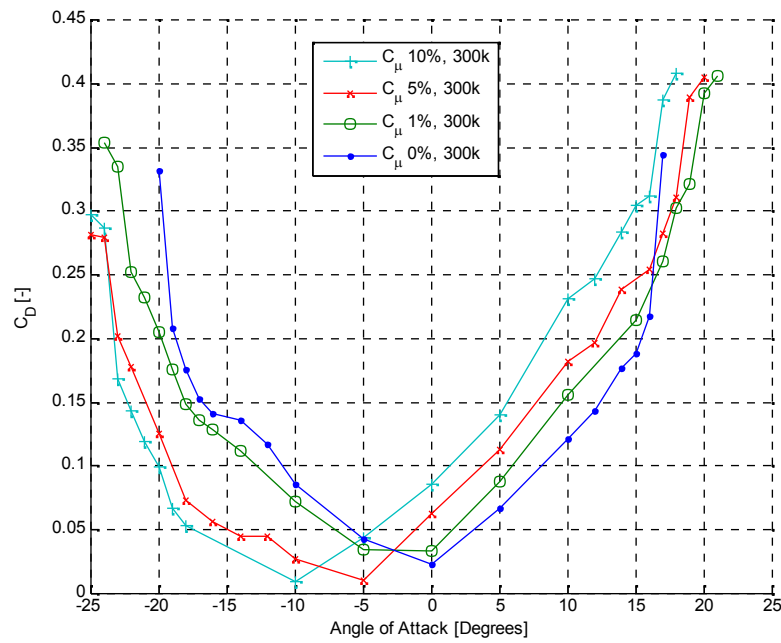
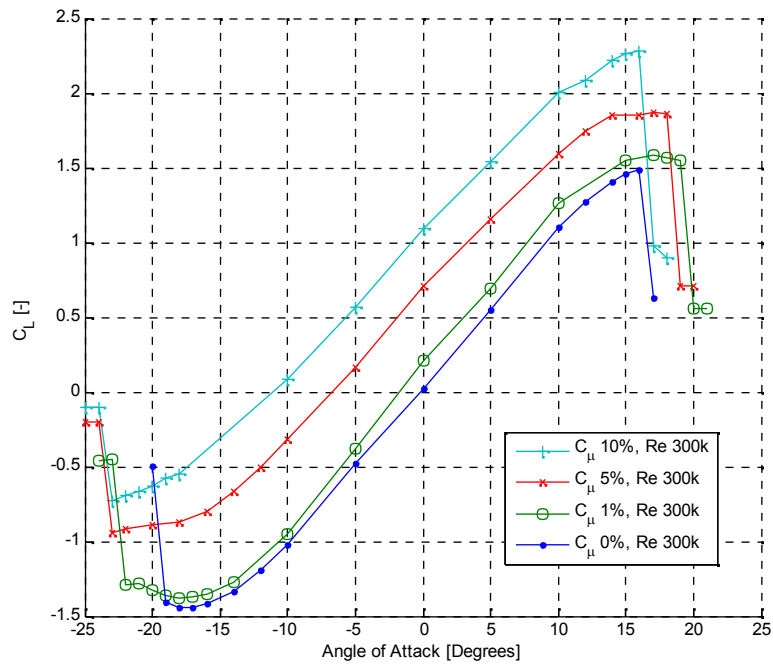
Figure 4-6 CC NACA 0018 Model Mounted in Wind Tunnel

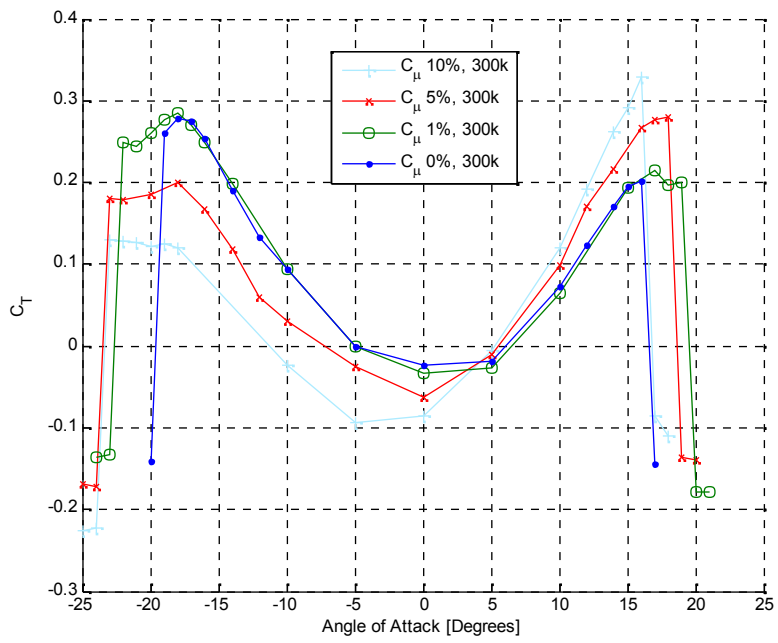
4.4 The CC-NACA0018 Rapid Prototype Model

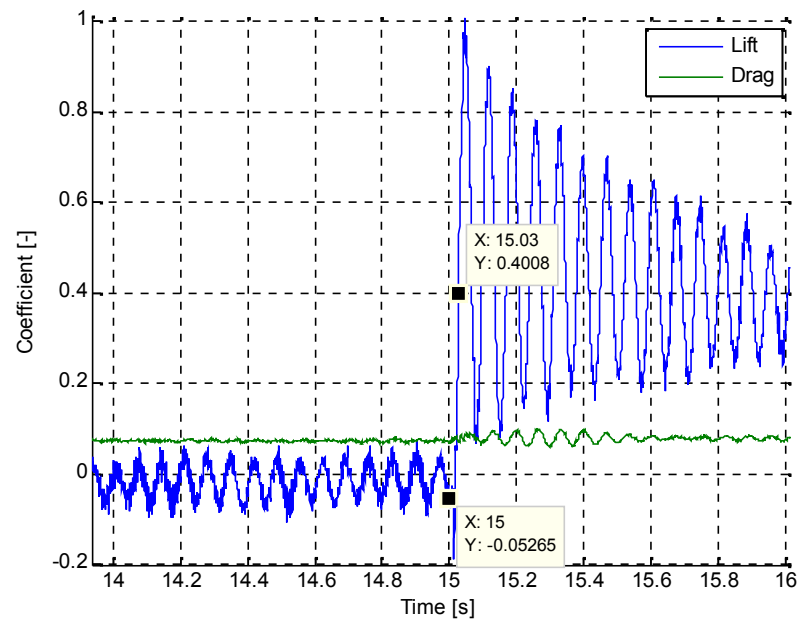
A second wind tunnel model was constructed to determine the response time of the internal valving. The cross section of this model was the same as the previous model, but the material used was plastic (printed by a 3D rapid prototyping machine) for the skin, ribs, and diffuser. Metal supports were

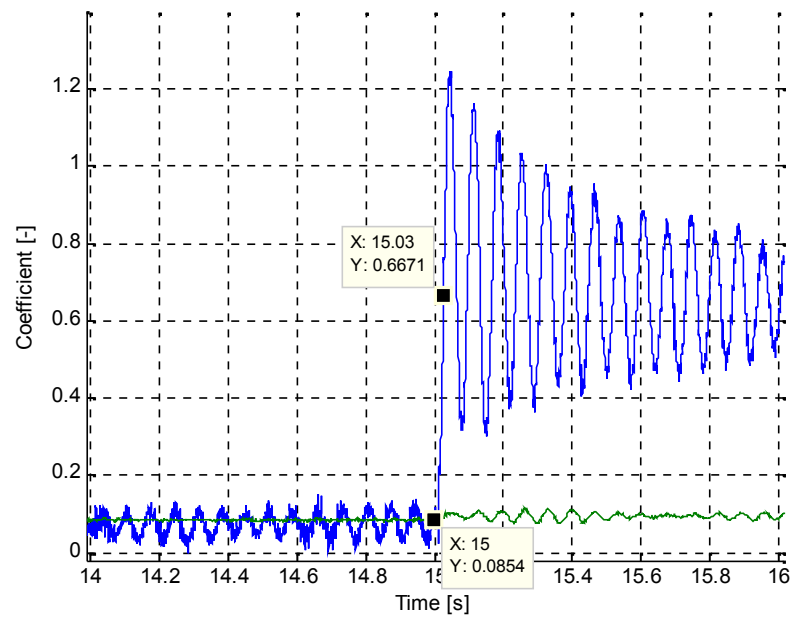


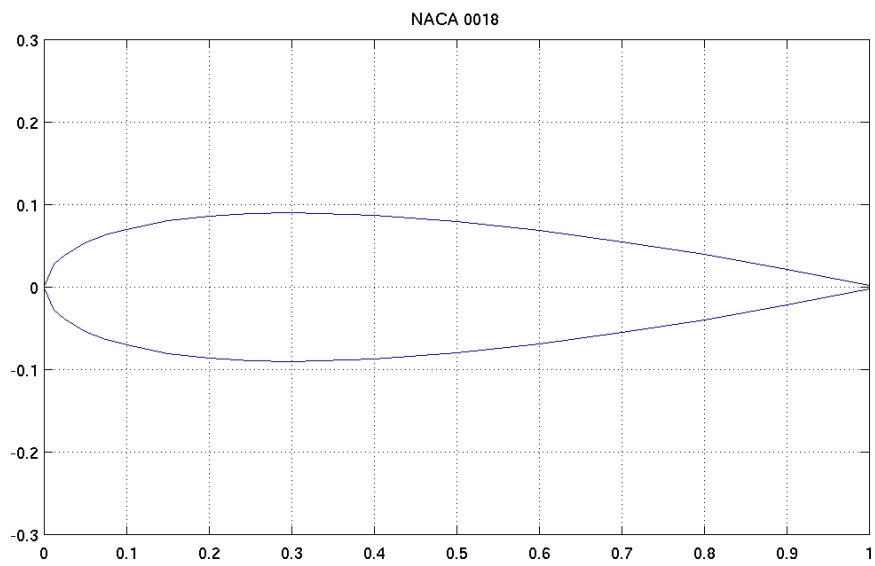
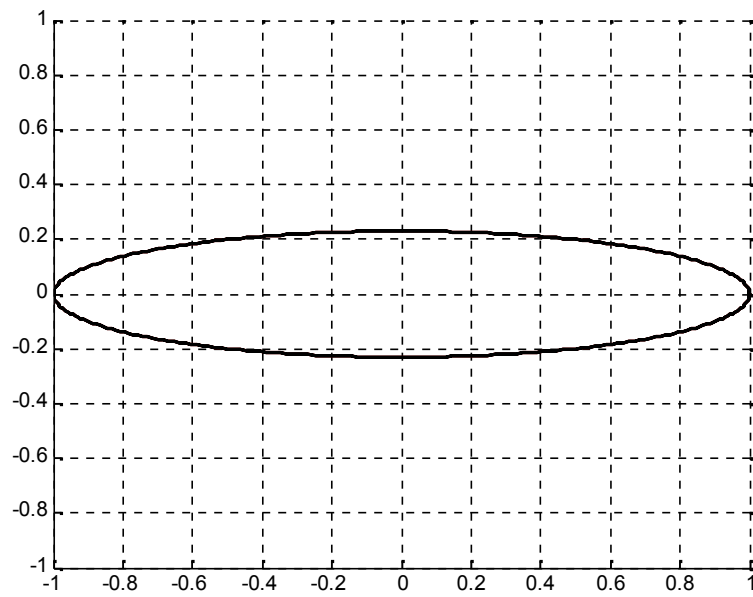


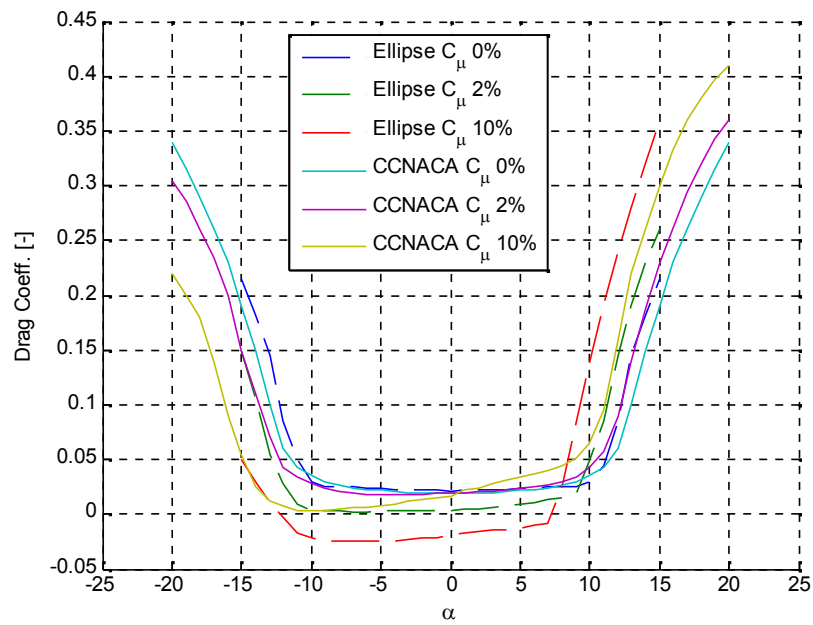
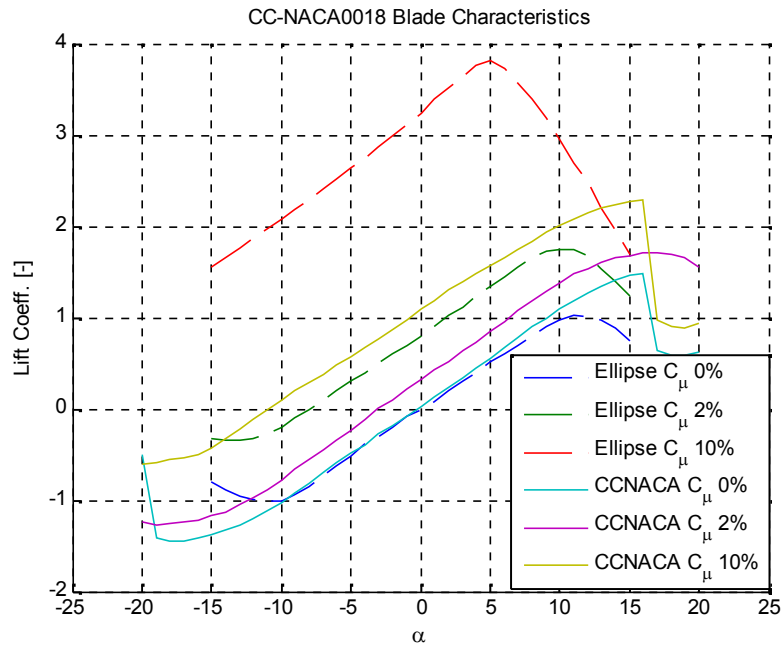


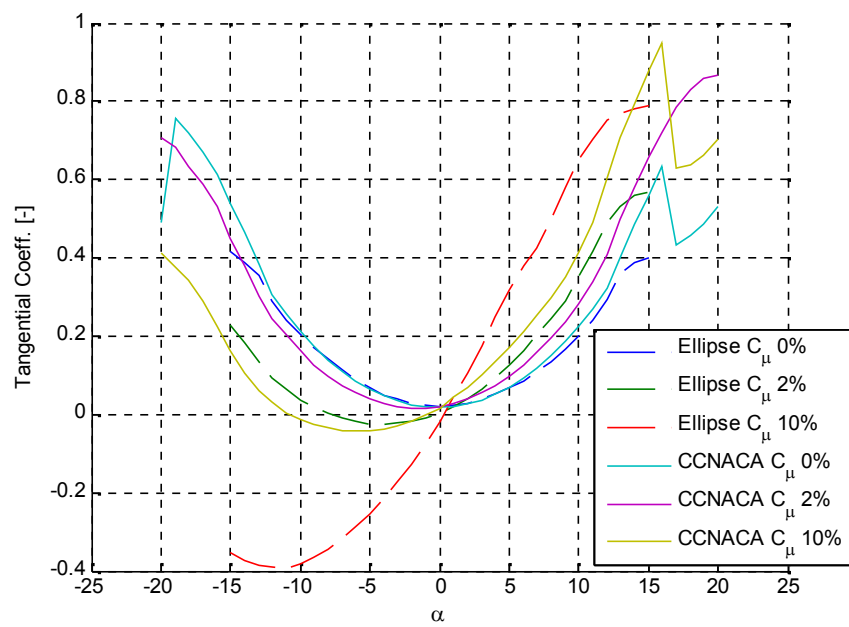


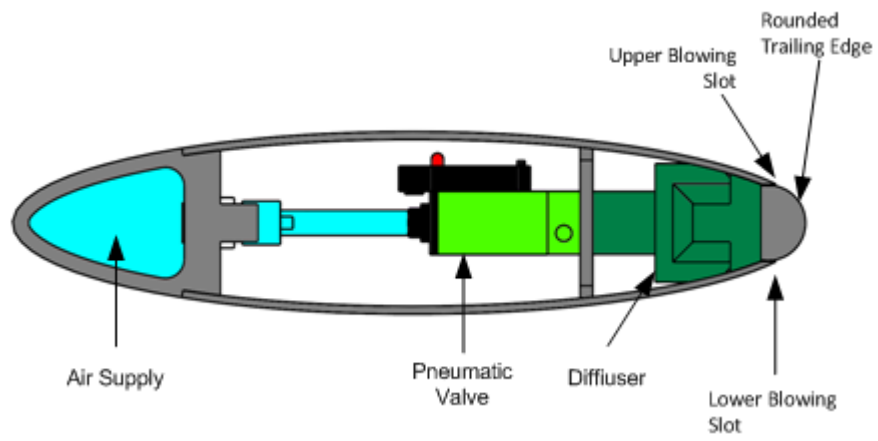












$$c_{\mu} = \frac{2hV_{jet}^2}{CV_b^2}.$$

The jet velocity based upon the blowing coefficient and blade velocity can be rearranged from equation (4.1) as

$$V_{jet} = \sqrt{\frac{c_{\mu} V_b^2 C}{2h}}. \quad (4.2)$$

The general form of the power required for a blowing jet is defined as

$$P_{jet} = \frac{1}{2} \rho_{jet} A_{jet} V_{jet}^3. \quad (4.3)$$

Substituting the jet velocity from (4.2) the power of the jet per unit span becomes

$$P_{jet} = \frac{1}{2} \rho_{jet} h \left\{ \frac{c_{\mu} V_b^2 C}{2h} \right\}^{\frac{3}{2}}. \quad (4.4)$$

The power required to generate the jet was defined by equation (4.4) with respect to the blowing coefficient. This could be used to evaluate net power extraction from the CCVAWT, but further investigation into non-steady blowing is required. Several pieces of work have shown a 50% or more decrease in mass flow using pulsed blowing. The pumping or blowing costs would greatly impact overall net energy capture and therefore an analysis should be performed when non-steady blowing power information is available for a particular blade valving implementation. (Lovato 1992, Cortelezzi 1994, Cagle 2002, Jones 2003, Heyes 2004, Liu 2004, Jones 2005, Goskel 2007 Liewkongsatoporn 2008)

4.9 CC-VAWT Blade Characteristics

Experimental lift and drag data for a CC-NACA0018 and literature data on a CC-Ellipse were compared for use in a CC-VAWT. Also included is the lift response time of the CC-NACA0018, which determines how many times the valves may be opened and closed during rotation of the turbine to effect performance. Although circulation control does increase lift of the CC-NACA0018 aerofoil, it is not

as efficient as the elliptical shape as found in literature. It is therefore recommended to utilize an elliptical shape for use in the CC-VAWT due to its higher efficiency when using circulation control.

Chapter 5. Solidity Matching Scheme Using Circulation Control for a VAWT

5.1 Introduction

The increased lift from circulation control is predicted to affect the energy capture of a VAWT. A logical scheme must be created to effectively use blowing for an increase in overall energy capture. Circulation control should follow a logical blowing scheme during rotation and for a given Tip Speed Ratio (TSR or λ). The goal of a control scheme for the CC-VAWT is to operate most efficiently at a particular, or design, wind speed without the use of circulation control. Blowing will then only be used when operating outside of the design optimum which reduces pumping energy costs. Several methods of blowing were analyzed and evaluated for their impact on performance and the control goals. [Equation Chapter 5 Section 1](#)

5.2 Constant and Variable Rotation Speed

The rotational speed of a wind turbine is either constant or variable. Variable rotational speed VAWT's have no clear advantage over a constant rotational speed VAWT because of the added costs of gearing or power processing or both. Mainly, the variable speed VAWT would use load and/or aerodynamic controls to operate at the maximum coefficient of performance for a given wind speed. A controller could, for a variable speed setup, change the speed to the TSR where the maximum C_p occurs and then try to hold steady at that point. As the wind speed changes, for a variable speed setup, the turbine will most likely be unable to maintain this C_p . Since the TSR is determined by both rotational

speed and wind speed the variable speed turbine would need to vary the rotational speed over coming rotor inertia to match the optimum TSR.

In some cases, such as turbines with large inertia, it may be impractical to vary the rotational speed at the same rate the wind speed changes. Constant speed turbines, on the other hand, could be driven by a power grid or other constant frequency power sources and will always be at a constant rotational rate. The only time the turbine will be out of sync with respect to the rotational rate is during startup or over-speed. Power required for startup, when compared to a long time averaged power capture, becomes insignificant. Constant speed grid-tied turbines would require grid power to effect startup and when the wind is unable to maintain the design speed. Because of these reasons, a constant speed grid tied turbine was selected for this study. Although a variable speed CC-VAWT may benefit from circulation control if a proper control scheme was developed.

5.3 Rotational Blowing Scheduling

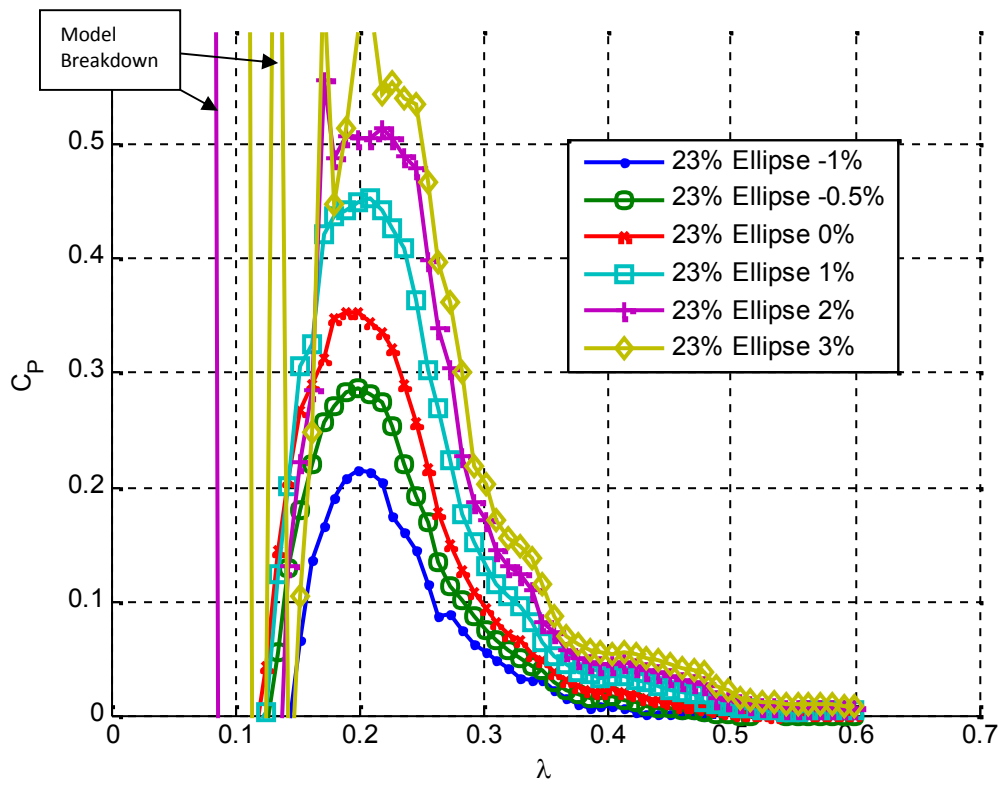
The main goal of a CC-VAWT controller is to determine the state of blowing during rotation for a tip speed ratio (TSR). The blowing may be at a constant or variable rate. A constant blowing rate means that the jet mass flow and velocity are fixed. Variable blowing is defined as a variable jet mass flow rate and/or velocity. The variable blowing rate would either maintain a constant blowing coefficient or follow a variable blowing coefficient. A constant blowing coefficient would require the mass flow and/or velocity to vary or match a known ratio depending upon the blade velocity at a particular point during rotation. The variable coefficient blowing would be used to follow a curve to create a desired performance curve.

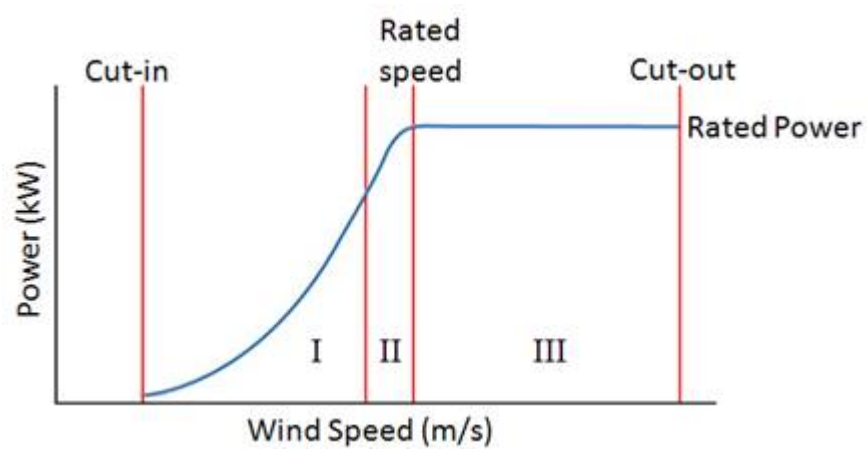
Preliminary simulations of the CC-VAWT with constant blowing rates and coefficients during rotation showed a relationship between a performance increase and the blowing coefficient, as shown

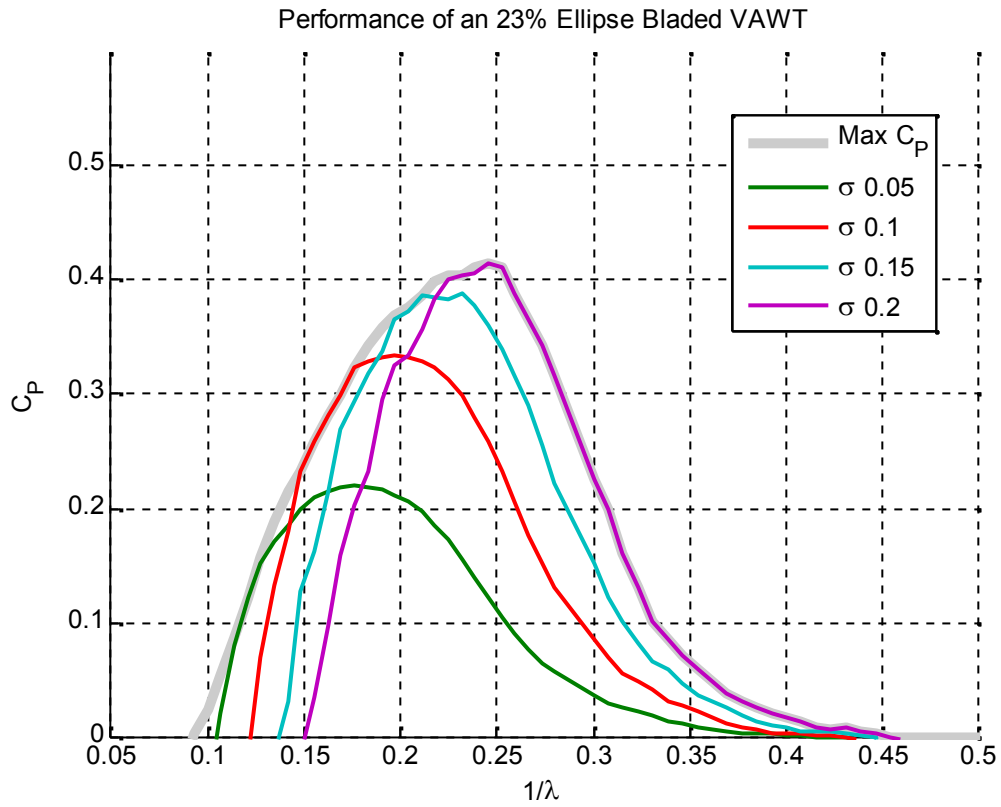
in Figure 5-1 for constant blowing coefficients. The x-axis of this power curve is represented by the inverse of the TSR,

$$\frac{1}{\lambda} = \frac{\omega R}{V_{\infty}}, \quad (5.1)$$

for a direct correlation to wind speed while still maintaining a non-dimensional relationship. Constant blowing coefficients, during rotation, greater than 2% and TSR more than 5 create instabilities in the numerical simulations used. From these simulations it is clear that constant blowing coefficient circulation control increases the performance, but it does very little to extend the performance over a wider TSR. In effect circulation control at a constant blowing coefficient increases the magnitude of the performance curve, but does not change the basic shape or make it cover more area. Circulation control increases the lift and in effect the tangential force, but this does not always increase overall performance or make the curve wider.







$$\sigma = \frac{N_b C}{R}$$

The performance of four different solidities is illustrated in Figure 5-3 where a solidity of 0.05 covers a wider range of TSR but is about half the peak of a solidity of 0.2. The maximum performance line of the solidity range from 0.01 to 0.2 is also shown at the max C_p curve to illustrate the performance a VAWT could cover as it adjusts solidity. A traditional fixed blade VAWT must stick to a single physical solidity and is limited to a single performance curve, but a circulation controlled VAWT has the ability to change its blade characteristics and produce more or less lift. Therefore it can change its virtual or apparent solidity and follow different power curves capturing both the highest peaks and wide range of TSRs.

Each solidity has its own performance curve and for each TSR there is only one solidity that has the maximum performance point. A circulation controlled VAWT could use blowing to increase the tangential force to match the maximum performance curve. Since the maximum performance is determined by solidity, matching the performance would effectively be changing to a different or virtual solidity. At each TSR to operate at the maximum possible coefficient of performance, an increase or decrease in the desired operating solidity may occur. Since it is known, from simulations, which solidity and TSR combinations produce the maximum performance at each TSR, this information could be used to create a control scheme that uses circulation control to vary the tangential force to produce the same performance as a different solidity. The tangential force and coefficient of performance vary during rotation, therefore the blowing must be variable during rotation.

The coefficient of performance is computed using the chord-to-radius ratio, TSR, and non-dimensional tangential force,

$$C_p = \frac{1}{2} \frac{c}{R} \lambda F_t, \quad (5.3)$$

where the tangential force per unit span can be expressed as

$$f_t = \frac{1}{2} \rho_\infty C c_t V_b^2 \quad (5.4)$$

and the non-dimensional tangential force is defined as

$$F_t = \frac{f_t}{\frac{1}{2} \rho_\infty C V_\infty^2} = c_t \left(\frac{V_b}{V_\infty} \right)^2 = c_t V_{rel}^2 \quad (5.5)$$

Combining these two equations, the coefficient of performance can be represented as

$$C_p = \frac{1}{2} \frac{c}{R} \lambda c_t V_{rel}^2 \quad (5.6)$$

Finally, the following expression, based upon equation (5.6),

$$\frac{1}{2} \frac{\sigma_o}{N_b} \lambda c_t [\alpha] V_{rel}^2 [\theta] = C_p [\sigma_M, \theta] \quad (5.7)$$

can be used to solve for the c_t required to match a performance level of a different solidity (σ_M) at a given initial solidity (σ_o).

The blade shape selected for this study was a 23% elliptical airfoil with a modified rounded trailing edge and a upper and lower surface blowing slot. The cross-section and internals of the CC-VAWT blade is shown Figure 5-4. The rounded trailing edge cuts out the elliptical edge, but according to (Abramson 1984) the base performance when not blowing is very similar to a non-modified ellipse.

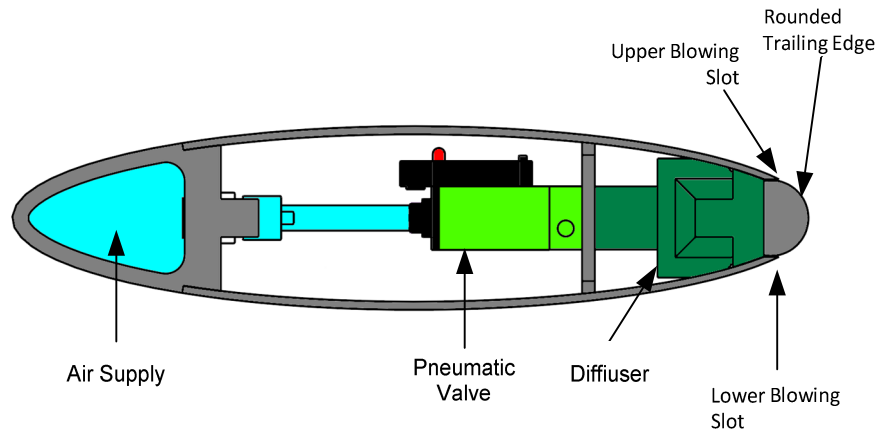
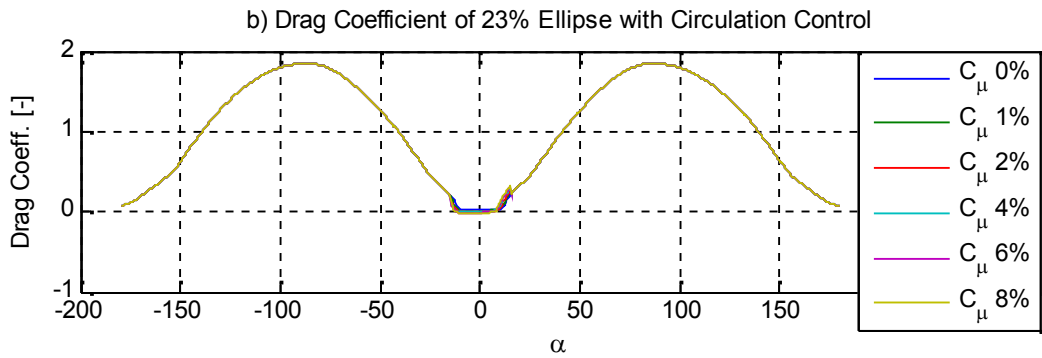
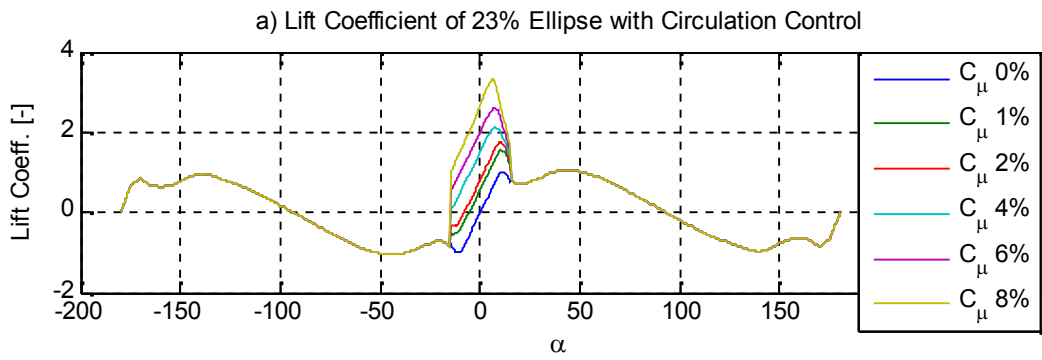
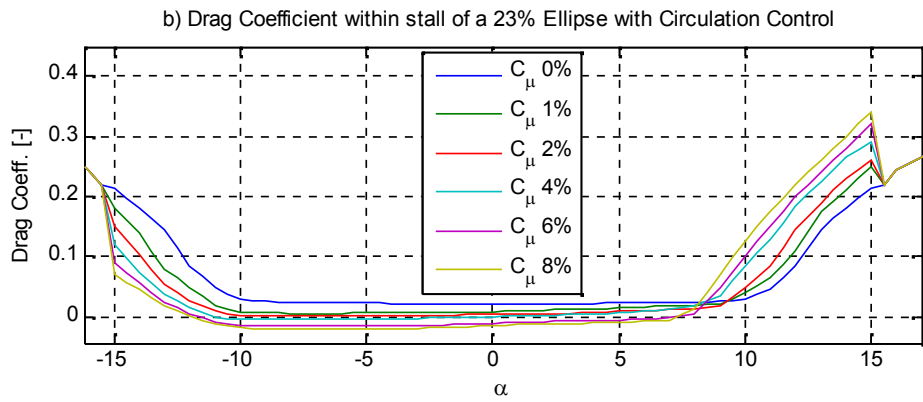
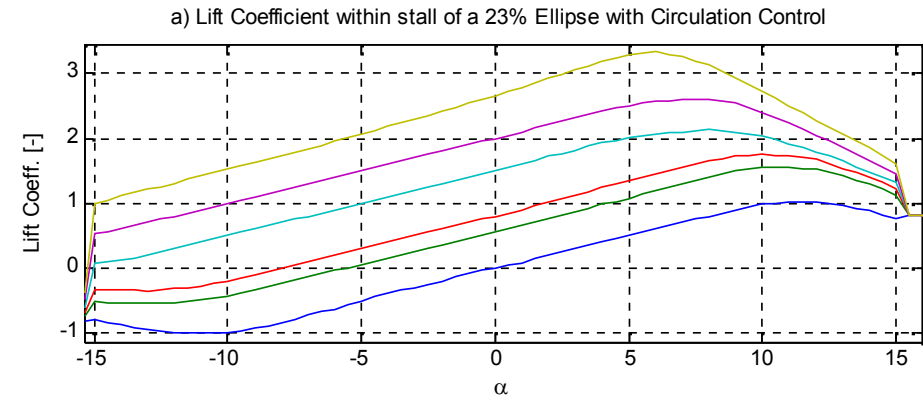
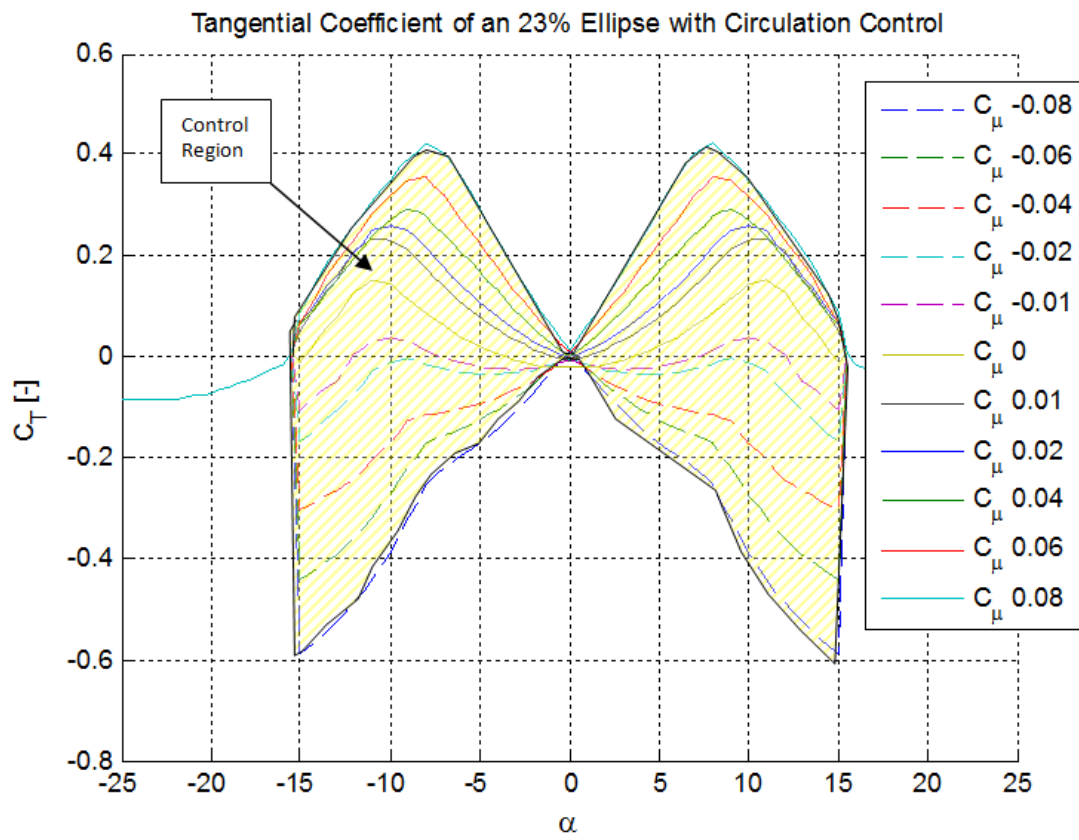


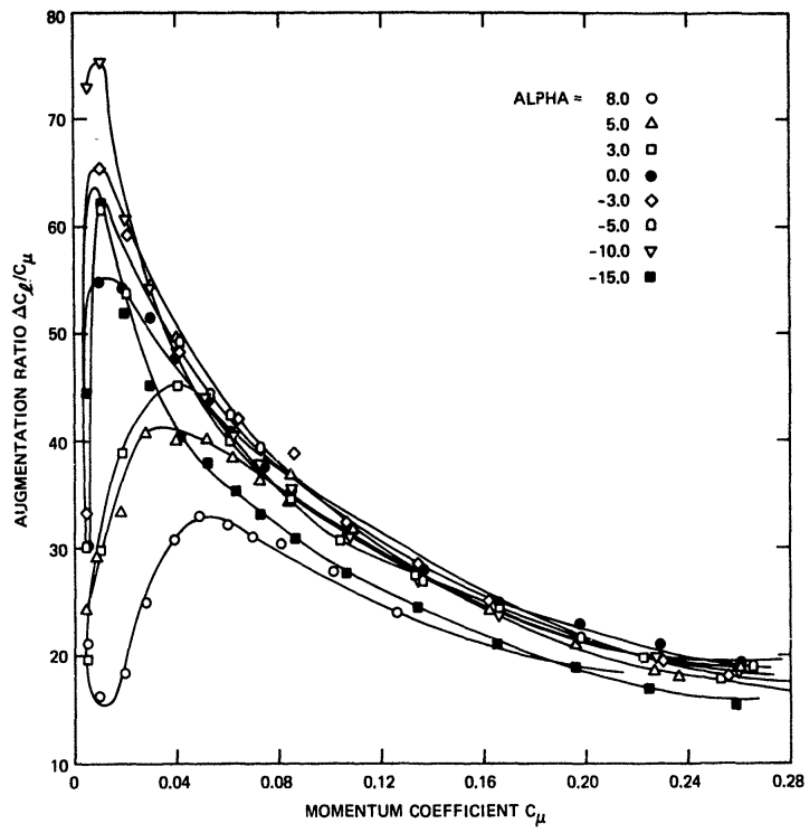
Figure 5-4 CC-VAWT Blade, 23% Ellipse

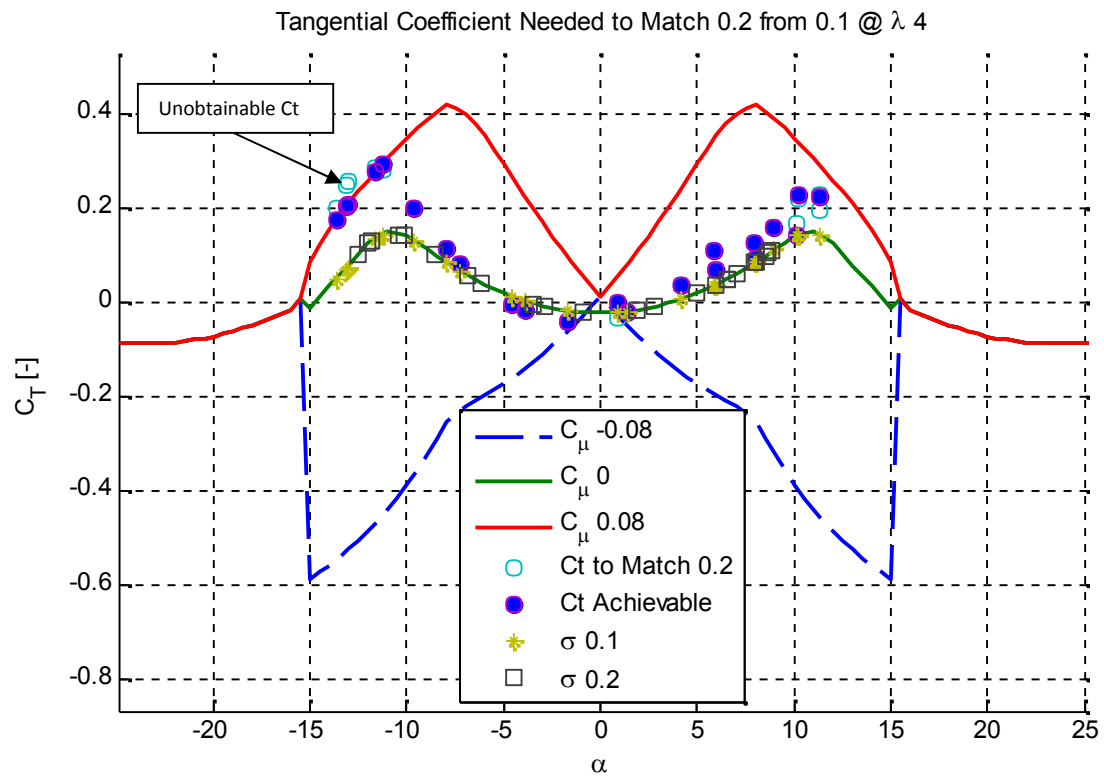
Coefficients of lift and drag used to describe the performance of the CC-VAWT blade with respect to blowing were developed by combining experimental data from various sources with similar airfoils (20%-25% ellipses) (Williams 1969, Englar 1972, Wood 1981, Abramson 1984, Rodman 1986, Franke 1988, Novak 1987, Abramson 2004, Alexander 2005, Wetzel 2009). The lift and drag with respect to the angle-of-attack is shown in Figure 5-5 for values within stall and in Figure 5-6 for the full 360 degrees. Lift and drag once the blade has stalled follows a NACA0018 (Sheldahl 1981). This was done because experimental data was only available for circulation controlled lift and drag data before stall and the analytical model requiring a full 360 degrees angle-of-attack data. The lift and drag is important to wind turbine performance but the tangential coefficient, a combination of lift and drag depending upon the angle-of-attack, can be directly correlated to performance.

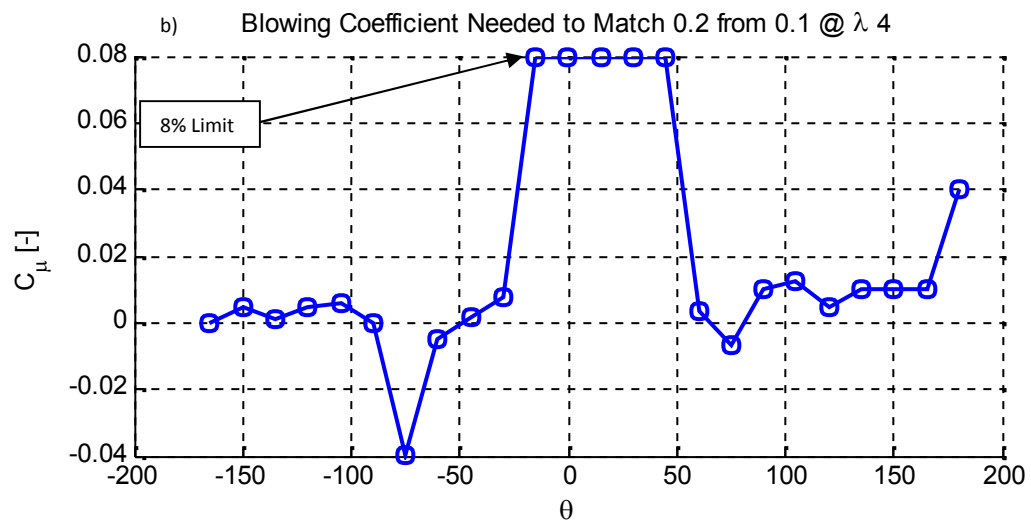
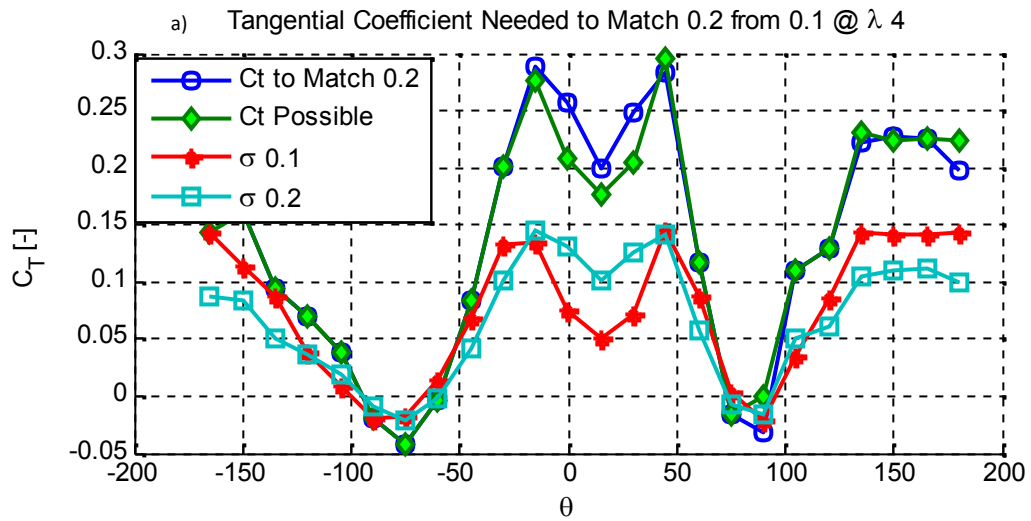


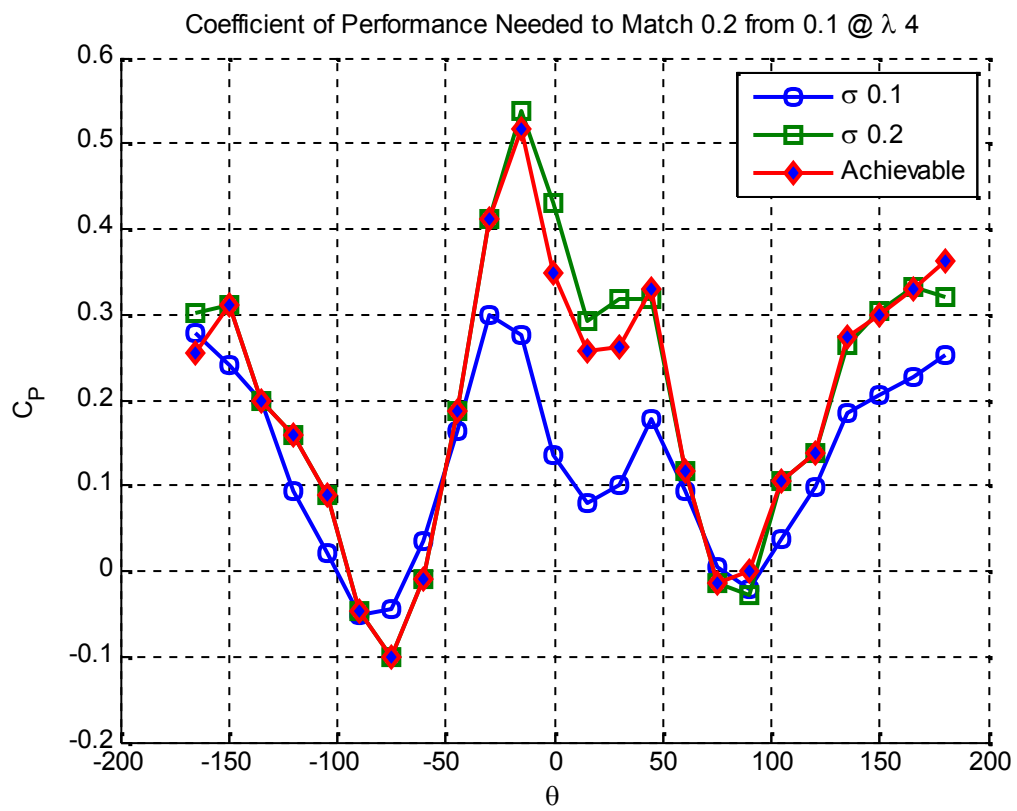
$$c_t = c_L \sin \alpha - c_D \cos \alpha.$$









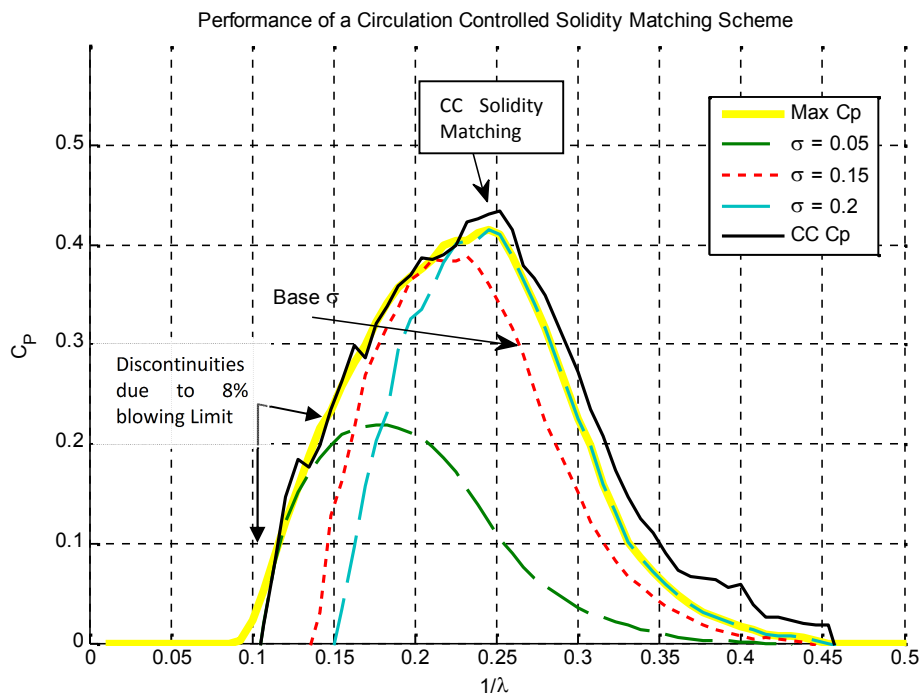
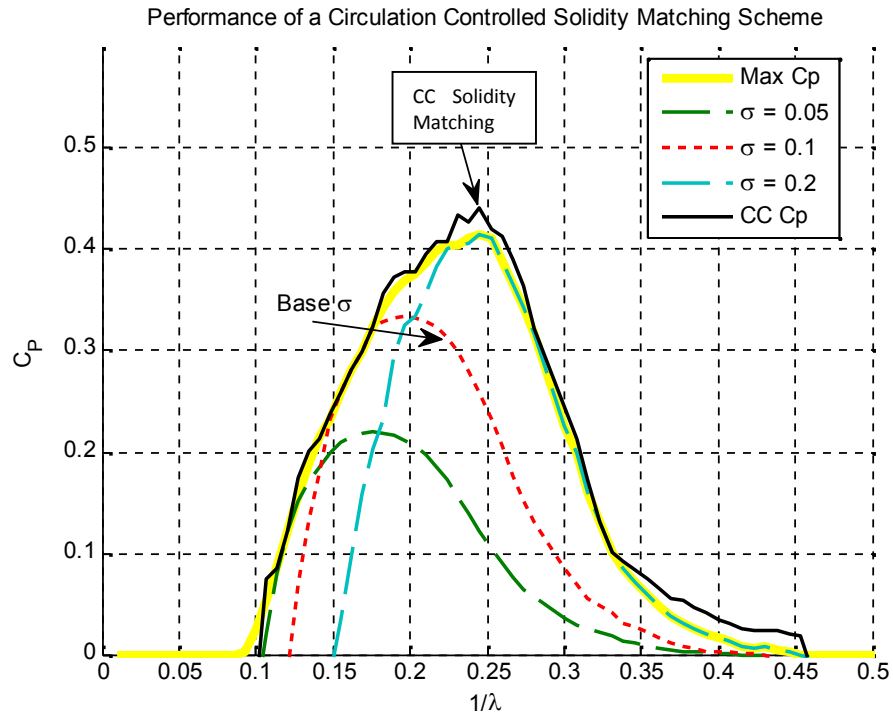


$C_{P_{\max}}$

- c. Outside of +/- 10 degrees (as stall is approached) a non-linear relationship between c_μ and c_t exists, a nearest interpolation is required.
- 4) Use the calculated c_μ and the airfoil data to determine the value of c_t that is achievable
- 5) Compute $C_{p-found}$ for the found c_t that is achievable

5.4 Evaluation of the Solidity Matching Scheme for Achieving a Maximum C_p

To test the solidity matching algorithm developed for maximum C_p , simulations using the 2D Vortex analytical model were conducted. The Vortex model was used with a controller that implemented the solidity matching algorithm to achieve a C_p at a desired solidity. The vortex model requires 2D airfoil coefficients in the form of lift and drag versus angle-of-attack to compute bound circulation and the tangential coefficient. At the point where the model uses the computed angle of attack to 'look-up' the lift and drag and compute c_t , the previously mentioned solidity matching steps are preformed with pre-computed C_p and solidity information. Results of this simulation can be seen in Figure 5-12 and Figure 5-13 where the initial solidity (σ_0) was 0.1 and 0.15 respectively along with three C_p curves for a solidity of (0.05, 0.1, and 0.2), the maximum C_p of the range of solidities (0.01 to 0.2), and the calculated C_p using the developed solidity matching scheme. At each TSR the solidity with the highest coefficient of performance was used in the controller for c_t matching. These figures demonstrate the ability of the solidity matching scheme to increase the area of the performance curve when using circulation control properly. The coefficient of performance is an indication of how well a turbine will perform, but a wind probability distribution indicates how much energy can be captured with respect to a span of time for a given installation site.



5.5 Energy Capture as Determined by a Wind Distribution

The wind probability distribution for a given average wind speed can generally be approximated by a Weibull distribution as defined by

$$f_w[V; \mu, k] = \frac{k}{\mu} \left(\frac{V}{\mu} \right)^{k-1} e^{-\left(\frac{V}{\mu} \right)^k} \quad (5.9)$$

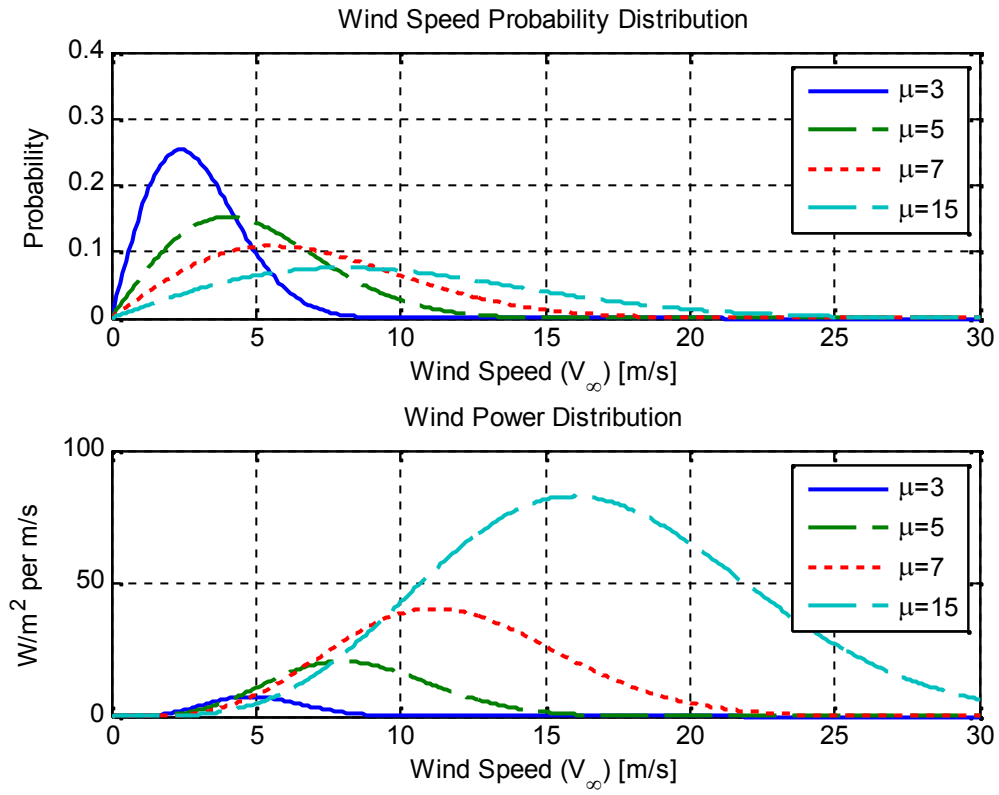
Using a Weibull distribution with a shape parameter (k) of $\frac{2\mu}{\sqrt{\pi}}$ with μ being the average wind speed, a Rayleigh distribution can be generated. This distribution determines the probability of a wind speed for a given average wind speed. Typically an installation site for wind turbines is evaluated and wind statistics are collected for up to a year. For the work performed, a Rayleigh distribution was assumed and was defined as a function of the Weibull distribution or

$$f_R[V; \mu] = f_w \left(V, 2, \frac{2\mu}{\sqrt{\pi}} \right) \quad (5.10)$$

The power probability density function of the wind can be computed from the Rayleigh distribution by

$$P = \frac{1}{2} \rho_{\infty} f_R[V, \mu] V^3 \quad (5.11)$$

Integrating the power probability distribution between two wind speeds gives the power available in that range of wind speeds. For a given average wind speed, the probability of wind and the power of that wind can be computed, shown by Figure 2-14 assuming an air density of $1.2 \frac{kg}{m^3}$. Notice that the peak of the power curves do not line up with the average wind speed nor do the most likely wind speeds. This wind power probability distribution may be used to evaluate a wind turbine's performs given an installation site's wind speed statistics.



$$P_{turbine} = \frac{1}{2} c_p \rho_\infty A V_\infty^3$$

$$A = 2bR$$

$$P_{turbine}^* = \frac{P_{turbine}}{b} = C_P \rho_\infty R V_\infty^3$$

Substitution of a Rayleigh distribution into the power capture equation of a wind turbine results in the following power distribution per unit span expression

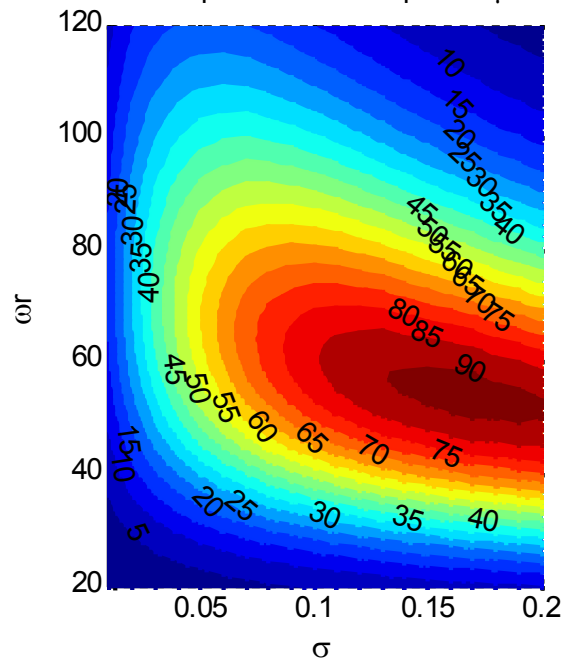
$$P_{turbineDist}^* = P_{turbine}^* f_R(V, \mu) = C_P \rho_\infty r V_\infty^3 f_R(V, \mu). \quad (5.15)$$

A wind turbines power capture at an installation site can be evaluated using equation (5.15) and the integral of this equation provides the overall power captured per unit span.

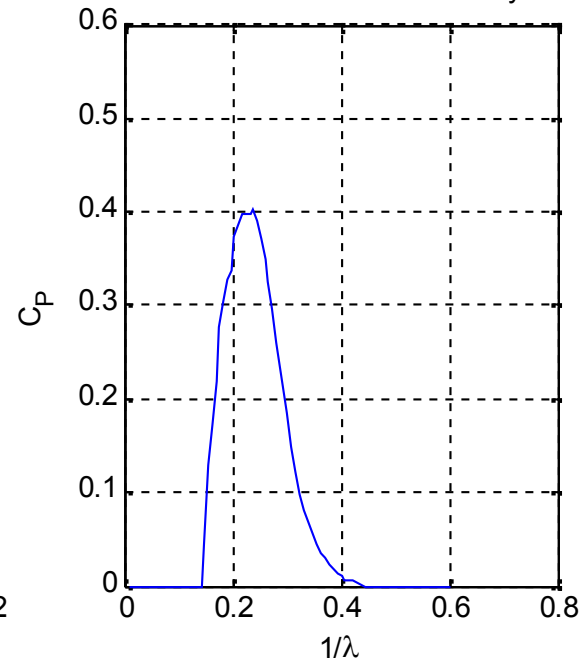
5.7 Evaluation and Performance for a Wind Distribution of the Solidity Matching Scheme

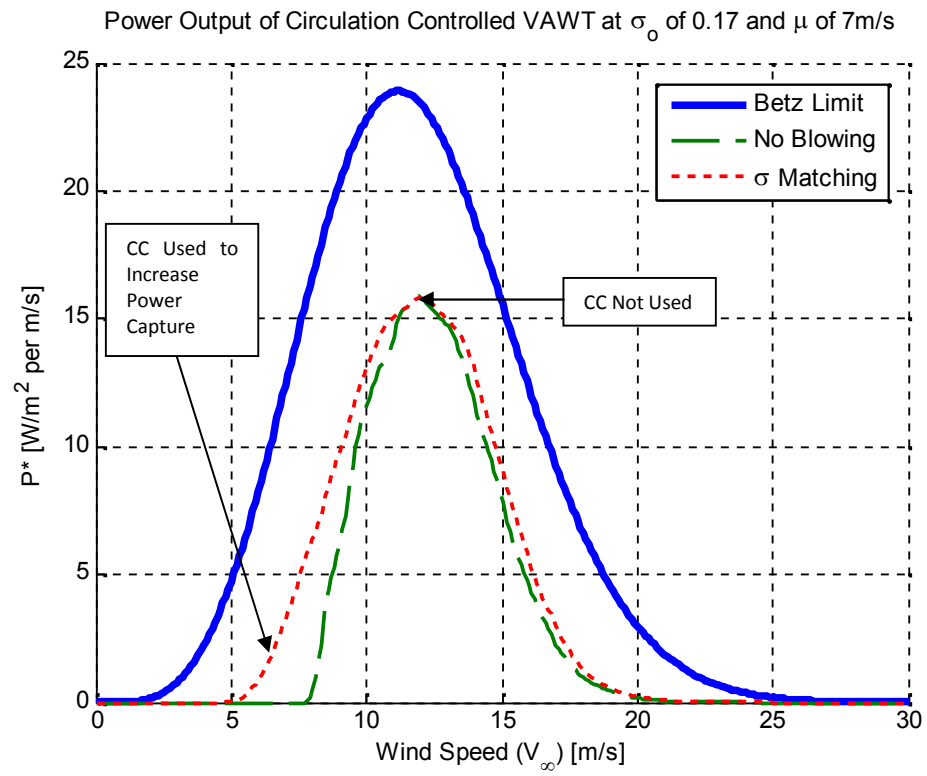
Evaluation of the power capture for the CC-VAWT was conducted using the vortex analytical model. First, the solidity of an VAWT with 23% elliptical blades (non-blowing case) were evaluated to determine which fixed rotational speed and solidity produced the highest power capture for a chosen average wind speed of 7 m/s. Second, using the optimum solidity and ωR found, the solidity matching scheme was simulated. The calculated power contour for a range of solidities and ωR is shown by Figure 5-15. Based upon this power contour, a solidity of 0.17 and ωR of 55 was found to produce the highest power capture at 91 W/m².

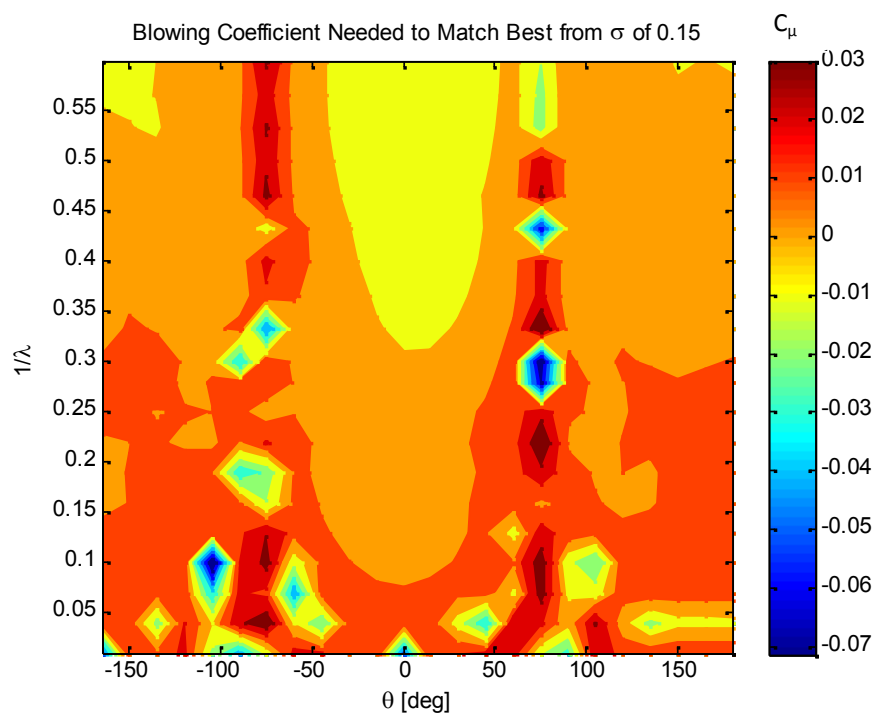
a) Power Capture of 23% Ellipse for $\mu = 7$

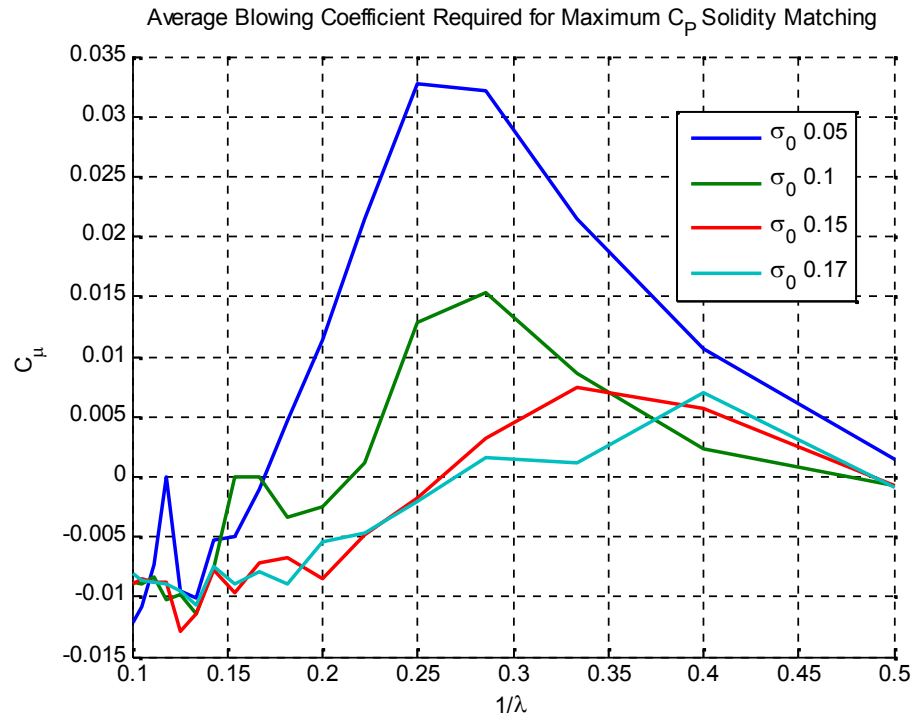


b) Coefficient of Performance for Solidity of 0.17

 ωR







Chapter 6. Experimental CC-VAWT Development

Based upon the preliminary and wind tunnel airfoil data, an experimental apparatus was designed to determine, in the future, the performance capabilities of a CC-VAWT. This apparatus contained a single circulation controlled blade and was a full size wind turbine. The solidity and TSR operating region of this design was determined from the preliminary analysis performed with the momentum model with constant blowing circulation control.

The developed circulation controlled blade of the CC-VAWT is shown without a surface skin in Figure 6-1, was constructed for aerodynamic load testing to verify the developed analytical model. The circulation control blade, an 0.2 m (8 in.) chord, contains 86 pneumatic valves, half for the upper and half for the lower surface slot. The CC-VAWT is straight bladed and was 4 m (13.3) ft in radius and 3 m (10 ft.) of blade span. The solidity of the CC-VAWT is 0.05 and the chord to radius ratio is 0.05. A counter weight, equivalent to the circulation controlled blade, was used to balance out the rotational load. Air is supplied by a compressor through an air coupler to a hollow shaft and support arms to the circulation controlled blade. In order to control the wind speed an 4.8 m, (16 ft.) propeller is located 6m (20 ft.) away from the turbine which can provide wind speeds up to m/s 12 m/s (26.8 mph).

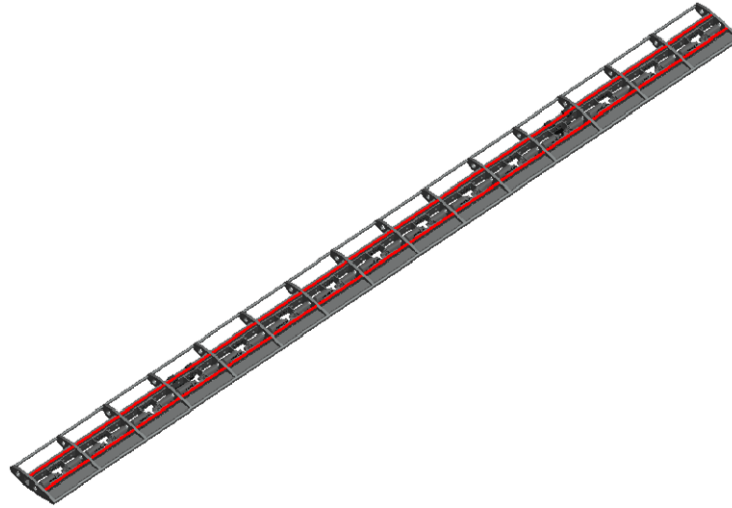


Figure 6-1 Circulation Control Turbine Blade (Transparent Skin)

The dimensions of the CC-VAWT experimental apparatus is shown in CAD drawing by Figure 6-2. The developed wind turbine shown in a similar orientation to the CAD drawing is pictured in Figure 6-3. The TSR may be variable and is dependent upon the rotational rate (0-100 RPM) and the capabilities of the wind simulator (0-15 m/s). The free stream velocity was controlled by a propeller placed 20 m away from the center of the turbine. The wind simulator, or propeller, can be seen with the CC-VAWT device fully constructed in Figure 6-4. A flow straightening device was used in-between the propeller and the turbine to reduce air swirling. The experimental setup of the wind speed simulator and the wind velocity profile can be found in (Kweder 2010).

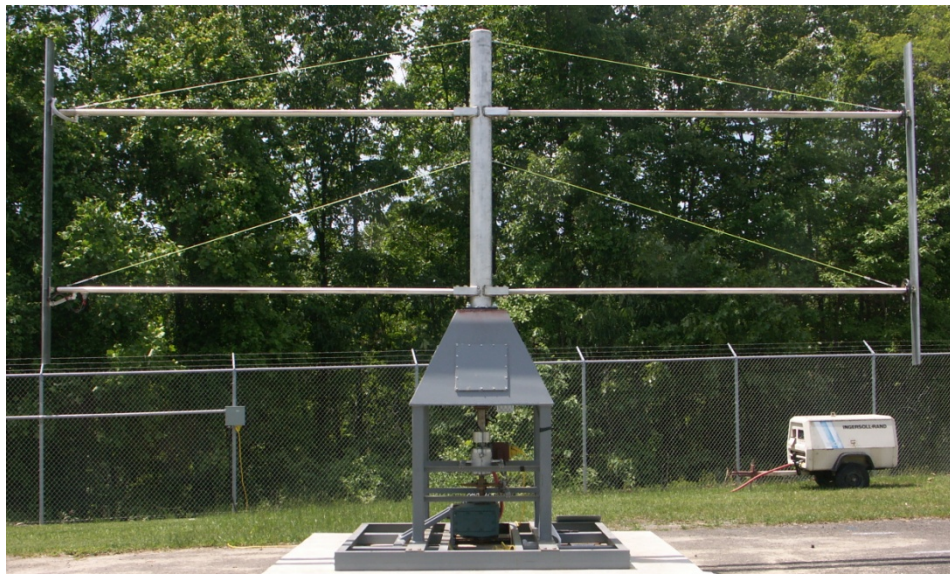
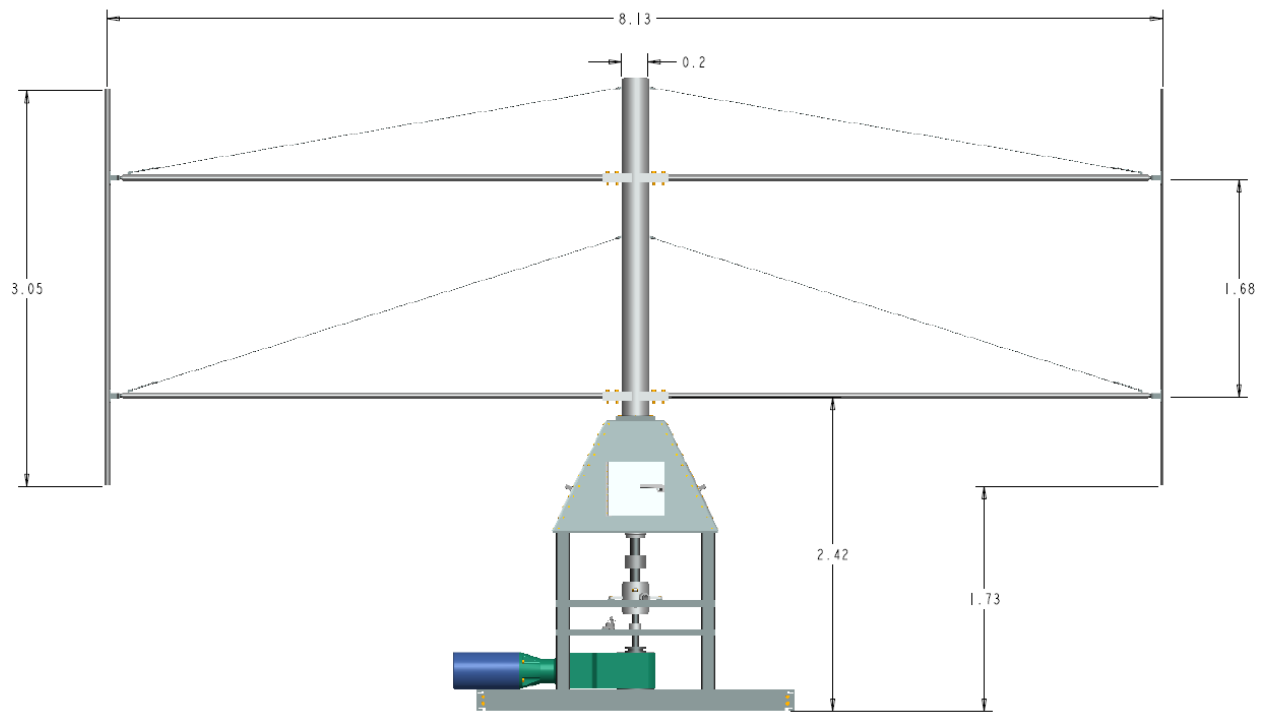




Figure 6-4 CC-VAWT and the Wind Simulator

6.1 Data Acquisition

Torque, free stream velocity, rotational position and speed, and atmospheric conditions will be recorded by a DAQ system. The rotational position and speed will be measured using an optical interrupter device with a keyed plate for a measurement accuracy of 1 degree (Pertl 2010). The free stream velocity will be measured with an ultrasonic anemometer (NovaLynx 200-7000). Atmospheric conditions, air temperature, atmospheric pressure, ambient wind speed and direction are going to be recorded during testing with a weather station (Oregon Scientific WMR968). Torque is planned to be recorded with the use of two shear gauges, each 180° apart from each other on the main shaft slightly above the main bearing.

The torque measured on the main shaft (τ) is related to the voltage ratio measured by connecting the two shear gauges into a full Wheatstone bridge. The relationship is represented as

$$\tau = \left[\frac{V_o}{V_i} \right] \left[\frac{\pi E (R^4 - R_i^4)}{4G_f R (1 + \nu)} \right], \quad (5.16)$$

where R is the radius of the main shaft, R_i is the thickness of the main shaft, E is Young's modulus, G_f is the gauge factor (Beer 2005).

The main shaft was 6061-T6 aluminum and is 0.2 m (8 in) in diameter. The value of Young's modulus (E) used was 68.9 GPa (10,000 ksi) and Poisson's Ratio (ν) was 0.330 for the material of the main shaft. The gauges used were type SGT-3H/350k-SY13 and have a gauge factor (G_f) of 2.02. The

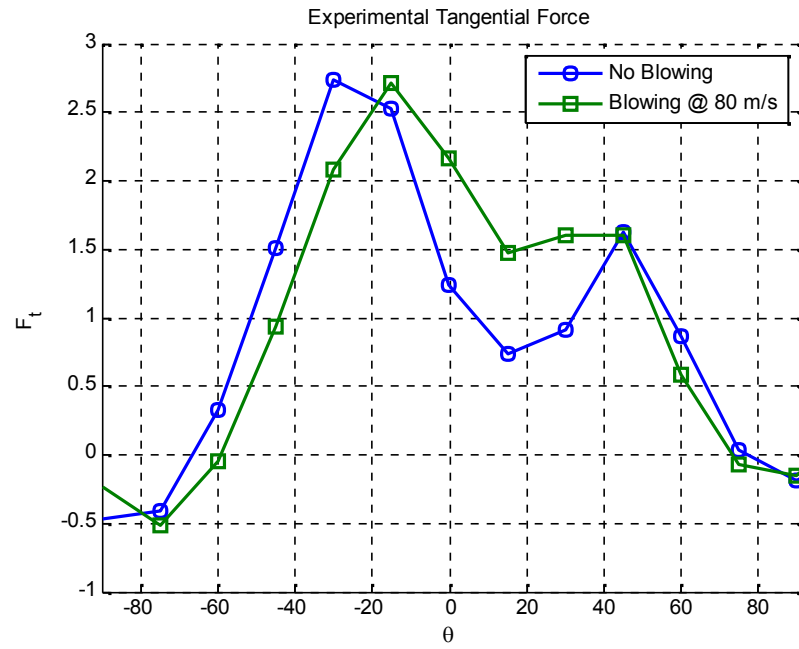
voltage from the Wheatstone bridge (V_o) was measured and during data reduction was divided by the supply voltage (V_i) and used in equation (5.16) with said parameters to determine the torque for the rotational positions from -90 to +90 degrees of rotation for every degree.

The torque force can be translated to a tangential force, for analytical model comparasion, by the following equation

$$F_t = \frac{\tau C}{\rho V_\infty^2} . \quad (5.17)$$

6.2 Tangential Force Predictions

The developed analytical model was used to predict the forces generated with constant blowing rate circulation control for the upwind portion of the VAWT's rotation. The upwind half of the turbine was only predicted because the aerodynamic interactions with the second blade of the CC-VAWT device is currently unknown. The predicted tangential force at a TSR of 4 without and with an 80 m/s jet with a 0.381 mm slot height is shown in Figure 6-5. The intent of this tangential prediction was to compare experimental results to the analytical model



Chapter 7. Conclusion

A circulation control blowing scheme for a constant rotational speed VAWT was developed by researching VAWT literature, implementing an analytical model of a VAWT, determining a blade shape, simulating the CC-VAWT, and performing an experiment with a CC-VAWT. The literature review indicated opportunity for an improvement in overall energy capture when using circulation control. A blade was determined by examining wind tunnel and literature airfoil performance data. Two analytical models were examined and implemented for performance predictions and simulations of a controller. Finally, a controller for an overall increase in energy capture was developed and simulated.

State-of-the-art wind turbine topics present in literature such as design methods, modeling techniques, control over wind turbines and power evaluation methods were reviewed. It was found that there are two commonly used and accepted methods of aerodynamically modeling a VAWT, the momentum model and the vortex model. Also found were simulations of a VAWT where the aerodynamics of the blade are changed by feathering. These simulations showed an increase in power production, but were unable to be implemented and tested due to structural issues. Also, found in the literature, were analytical modeling techniques for a VAWT and that indicated an increase in energy capture using circulation control.

Two analytical aerodynamic models of a VAWT were explored for performance predictions and controller development. The momentum model, a basic aerodynamic model, was used for an initial analysis of a VAWT. An initial analysis with the momentum model was performed with circulation control to determine the basic performance of circulation control using the preliminary airfoil data. The vortex model was used for further analysis and controller development based upon an circulation controlled elliptical blade. This vortex model was then used for controller development because the momentum model had breakdown issues or inabilities to provide a solution at high solidities and

blowing rates. Investigations into solidities up to 0.2 with blowing were necessary to better understand the power curves. The VAWT forces of lift, drag, and torque and the coefficient of performance, within TSR and solidity regions, were predicted by these models. The model was then used for development of a blowing control scheme for maximum energy capture of a CC-VAWT.

It was found that the solidity factor, a design parameter, has the largest impact on performance, but a turbine can only have a single physical solidity. A controller was then developed to use the enhancements of circulation control to mimic a virtual solidity change given an initial or physical solidity for constant rotational speed. Simulations of this controller over a wind speed range using a wind probability distribution indicated an approximate 22% increase in energy capture when using the control scheme.

7.1 Research Conclusion

A circulation controlled vertical axis wind turbine (CC-VAWT) was modeled and simulated. The developed analytical model was used to design a control scheme for the blowing slots to maximize the energy capture by increasing the area of the power curve. Circulation control was found to impact the power curve of a CC-VAWT which effectively changed its virtual solidity. Employing this technology, maximum energy capture was achieved while following the developed control scheme during simulations.

7.2 Recommendations

Although a blowing slot control scheme was developed there are still more topics and methods to explore. Dynamic stall was not included in this work because of the unknown effect that circulation control has on stall. A numerical investigation or an experiment could provide insight into dynamic stall of a circulation control blade, when operating under steady state or dynamic blowing conditions. Flow

curvature should also be investigated because of its impact on circulation control and the desired solidity selected for operation. The jet power should be used to evaluate net power extraction from the CCVAWT, but further investigation into non-steady blowing is required. Several pieces of work have shown a 50% or more decrease in mass flow using pulsed blowing with the same or similar lifting performance. The pumping or blowing costs would greatly impact overall net energy capture and therefore an analysis should be performed when non-steady blowing power information is available for a particular blade valving implementation. The implementation of the control scheme requires a variable blowing rate. Pneumatic valves would be required, either through pulsed blowing or something else, that would allow for a variable blowing rate to be used. A constant speed CC-VAWT was used to develop a blowing slot controller, but there may also be benefits to using a variable speed setup. In summary the following topics should be explored.

- Variable Speed VAWT
- Cut-in Speed
- Cut-out Speed
- Jet Pumping Power Analysis with Pulsed Blowing
- Further Blade Shape Investigation
- Variable Flow Rate Valves
- Circulation Control Dynamic Stall
- Flow Curvature

References

- (Anderson 2007) Anderson Jr., John D., Fundamentals of Aerodynamics 4th ed., Singapore China, McGraw Hill, 2007.
- (Abramson 1975) Abramson, J., "Two-Dimensional Subsonic Wind Tunnel Evaluation of a 20-Percent-Thick Circulation Control Airfoil," David W. Taylor NSRDC-ASED 311, 1975.
- (Abramson 1984) Abramson, J., "Low-Speed Characteristics of a Circulation Control Airfoil with Aft Camber and A Spiral Trailing Edge," David W. Taylor DTNSRDC/ASED-84/07, 1984.
- (Abramson 2004) Abramson, J., "Characteristics of a Cambered Circulation Control Airfoil Having Both Upper and Lower Surface Trailing Edge Slots," NSWCCD-50-TR-2004/030, 2004.
- (Alexander 2005) Alexander, G., Johnson, S., Florance, J., Keller, D., "Trailing Edge Blowing on a Two-Dimensional Six-Percent Thick Elliptical Circulation Control Airfoil Up to Transonic Conditions," NASA Technical Report, Langley Research Center, Hampton, Virginia, NASA/TM-2005-213545, 2005.
- (Angle 2008) Angle, G., Aerodynamic Benefits of Near-Surface-Actuated Circulation Control Blowing Slots for Rotorcraft Use, Ph.D. Dissertation, West Virginia University, Morgantown, WV, 26508, 2008.
- (Beer 2001) Beer, F., Johnston, E., DeWolf, J., Mechanics of Materials, McGraw-Hill, New York, NY., ISBN 0-07-365935-5, 2001.
- (Bianchi 2007) Bianchi, F., Battista, H., Mantz, R., Wind Turbine Control Systems, Springer-Verlag London, 2007 ISBN 1846284929.
- (Brulle 1977) Drulle, R. "Giromill Wind Tunnel Tests and Analysis," Proceedings from the 3rd Biennial Conference Workshop on Energy Conversion Systems, Washington, DC, Sept. 19-21, pp 775-783, 1977.
- (Cardona 1984) Cardone, J., "Flow Curvature and Dynamic Stall Simulated with an Aerodynamic Free-Vortex Model for VAWT," Wind Engineering, Vol. 8, 1984, pp. 135-143.
- (Drees 1978) Drees, H., "ISES: A Two-Dimensional Viscous Aerodynamic Design and Analysis Code," AIAA Paper 87-0424, Jan. 1978.
- (Dreier 2007) Dreier, M., "Introduction to Helicopter and Tiltrotor Flight Simulation," AIAA Education Series, 2007, ISBN 1-56347-873-0.
- (Duquette 2003) Duquette, M., Visser, K., "Numerical Implications of Solidity and Blade Number on Rotor Performance of Horizontal-Axis Wind Turbines," Journal of Solar Energy Engineering, Vol. 125, Issue 4, 2003, pg 425-432.
- (Englar 1972) Englar, R. J. "Two-Dimensional Subsonic Wind Tunnel Tests of Two 15-Percent Thick Circulation Control Airfoils," *Naval Ship Research and Development Center*. Technical Note AL-211, Washington, 1972.

- (Englar 1975) Englar, R., "Experimental Investigation of the High Velocity Coanda Wall Jet Applied to Bluff Trailing Edge Circulation Control Airfoils," September 1975, Naval Research and Development Center Report 4708, 1972.
- (Fanucci 1976) Fanucci, J., Walters, R., "Innovative Wind Machines: The Theoretical Performances of a Vertical Axis Wind Turbine," In Proc. Of the VAWT Technology Workshop, Sandia Lab. Report SAND 76-5586, 1976.
- (Franke 1988) Franke, M. E. and J. K. Harvell. "Wind Tunnel Studies of Circulation Control Elliptical Airfoils." Air Force Institute of Technology (Report N88-17598), Wright-Patterson AFB, OH, 1988.
- (Friedman 2007) Friedman, Chen, Arieli, Rimon, Levy, Yuval, "Numerical Investigation of Unsteady Circulation Control," AIAA Technical Paper Series, Paper No. 2007-3918, 2007.
- (Gormont 1973) Gormont, R. "A Mathematical Model of Unsteady Aerodynamics and Radial Flow for Applications to Helicopters," Technical Report DAAJO2-71-C-0045, US Army Air Mobility Research & Development Lab Vertol Div. 1973.
- (Hansen 2004) Hansen, M. H., Gaunaa, M., Madsen, H. A., "A Beddoes-Leishman Type Dynamic Stall Model In State-Space and Indicial Formulations," Riso National Laboratory, Roskilde, Denmark, Riso-R-1354(EN), 2004.
- (Islam 2006) Islam, M., Ting, D., Fartaj, A., "Aerodynamic Models for Darrieus-type straight-bladed Vertical Axis Wind Turbines," Renewable and Sustainable Energy, Vol. 12, Issue 4, May 2008 pp 1087-1109.
- (Johnson 1970) Johnson, W. (1970, May). The Response and Airloading of Helicopter Rotor Blades due to Dynamic Stall. Technical Report TR 130-1, MIT Aeroelastic and Structures Research Lab, 1970.
- (Kirke 1998) Kirke, B. "Evaluation Of Self-Starting Vertical Axis Wind Turbines For Stand-Alone Applications," Ph. D. thesis, Griffith University, 1998.
- (Kweder 2010) Kweder, J., Smith, J., "Facility for Rotor Testing Velocity Field," *Unpublished*.
- (Leishman 1990) Leshman, J., "Dynamic stall experiments on the NACA 23012 aerofoil," Experiments in Fluids, Vol. 9, Issue 1-2, 1990, pp 49-58.
- (Leishman 2002) Leshman, J., "Challenges in modelling the unsteady aerodynamics of wind Turbines," AIAA 2002-0037, 21st ASME Wind Energy Symposium 2002.
- (Leishman 2006) Leishman, J. G. "Principles of Helicopter Aerodynamics," Cambridge University Press, 2006, ISBN 0521858607.
- (Larsen 1975) Larsen, H., " Summary Of Vortex Theory Of The Cycogiro. In Proc. 2nd national Conference on Wind Engineering Research," June 22-25, pp. Vol. 8, 1975, Issue 1.
- (Larsen 2007) Larsen, J., Neilsen, S., Krenk, S., "Dynamic stall model for wind turbine airfoils," Journal of Fluids and Structures, Vol. 23, 2007, pp 969-982.
- (Loth 1976) Loth, J., Chandra, Subrato, Gibbs, E., "Inlet and Internal Aerodynamics of VTOL Thrust Augmenter," Final Report, Office of Naval Research Contract N00014-75-C-0324, 1976.

- (Loth 1983) Loth, J., McCoy, H., "Optimization of Darrieus Turbines with an Upwind and Downwind Momentum Model," *Journal of Energy*, Vol. 7, No. 4 July-Aug. 1983, pp 313-318.
- (McConnell 1979) McConnell, R., "Giromill Overview," *Wind Energy Innovative Systems Conference*, Colorado, May 23-25, 1979.
- (McIver 1996) McIver, A., Holmes, D., Freere, P., "Optimal Control of a Variable Speed Wind Turbine Under Dynamic Wind Conditions," *Industry Applications Conference*, 1996. Thirty-First IAS Annual Meeting, IAS '96, Conference Record of the 1996 IEEE, Vol. 3, pp 1692-1698.
- (Migliore 1978) Migliore, P., A Free-Vortex Model with Numerical Solution for the Unsteady Lifting Characteristics of Straight Bladed Darrieus Wind Turbines, Ph.D. Dissertation, West Virginia University, Morgantown, West Virginia, 1978.
- (Monster 2008) "How to Build a Wind Turbine," <http://www.monsterguide.net/how-to-build-a-wind-turbine.shtml>, 2008.
- (Noll 1982) Noll, R., Ham, N., "Effects Of Dynamic Stall On SWECS," *Journal of Solar Energy Engineering* Vol. 104, 1982, pp 96-101.
- (Novak 1987) Novak, C. J., Cornelius, K. C., and R. K. Roads. "Experimental Investigation of the Circular Wall Jet on a Circulation Control Airfoil." AIAA-87-0155, 1987.
- (NVIDIA CUDA 2009) "CUDA Zone," http://www.nvidia.com/object/cuda_home_new.html, 2009.
- (Panther 2010) Panther, C., Williams, K., Wilhelm, J., Smith, J., "Experimental Testing of a Wind Tunnel Model for use in a Vertical Axis Wind Turbine," ASME IMECE2010-38910, 2010, *Unpublished*.
- (Paraschivoiu 1982) Paraschivoiu, I., "Aerodynamic Loads and Rotor Performance for the Darrieus Wind Turbines," AIAA 2nd Terrestrial Energy Systems Conference, Dec. 1-3 1981, AIAA-82-2582.
- (Paraschivoiu 1988) Paraschivoiu, I., Azeddine, A., "Aerodynamic Analysis of the Darrieus Wind Turbines Including Dynamic-Stall Effects," *Journal of Propulsion and Power*, Vol. 4, No. 5, pp 472-277.
- (Paraschivoiu 2002) Paraschivoiu, I., Saeed, F., Desobry, V., "Prediction Capabilities in Vertical-Axis Turbine Aerodynamics," *The World Wind Energy Conference and Exhibition*, Berlin, Germany, 2-6 July 2002.
- (Pawsey 2002) Pawsey, N., Development and Evaluation of Passive Variable-Pitch Vertical Axis Wind Turbines, Ph.D. Dissertation, School of Mechanical and Manufacturing Engineering, The University of New South Wales, 2002.
- (Padfield 1996) Padfield, G., *Helicopter Flight Dynamics: The Theory and Application of Flying Qualities and Simulation Modeling*, AIAA, 1996, ISBN 1563472058.
- (Rodman 1986) Rodman L., Wood, N., "Verification of Performance Results for a Low-Speed 15% Elliptical Circulation Control Airfoil," *Joint Institute for Aeronautics and Acoustics*, JIAA RT-56, NASA Ames Research Center, 1986.
- (Sedighzadeh 2008) Sedighzadeh, M., Rezazadeh, A., "Adaptive PID Controller based on Reinforcement Learning for Wind Turbine Control," *Proceedings of World Academy of Science, Engineering, and Technology*, Vol. 27, Feb. 2008.

- (Schlichting 1968) Schlichting, H. *Boundary-Layer Theory (6th Edition)*. McGraw-Hill Book Company, pp. 709-14, 1968.
- (Sharpe 1990) Sharpe, D., *Wind Turbine Aerodynamics*, Prentice Hall, Cambridge, 1990.
- (Sheldahl 1980) Sheldahl, R., Klimas, P., Feltz, L., "Aerodynamic Performance of a 5-Metre-Diameter Darrieus Turbine With Extruded Aluminum NACA-0015 Blades," Sandia National Laboratories, SAND80-0179, 1980.
- (Spera 1994) Spera, D., *Wind Turbine Technology*, ASME Press, 1994, ISBN 0791812057.
- (Staelens 2003) Staelens, Y., Saeed, F., Paraschivoiu, I., "A Straigh-Bladed Variable-Pitch VAWT Concept for Improved Power Generation," 41st Aerospace Sciences Meeting and Exhibit, Jan 6-9, Reno, Nevada 2003.
- (Strickland 1975) Strickland, J., "The Darrieus Turbine: A Performance Prediction Model Using Multiple Streamtubes," Sandia National Labs Report SAND75-0431, 1975.
- (Strickland 1980) Strickland, J., Webster, B., Nguyen, T., "A Vortex Model of the Darrieus Turbine: An Analytical and Experimental Study," Sandia National Labs Report SAND79-7058, 1980.
- (Strickland 1986) Strickland, J., Graham, G., "Dynamic Stall Inception Correlation For Airfoils Undergoing Constant Pitch Rate Motions," AIAA Journal Vol. 24, Issue 4, 1986, pp 678–680.
- (Templin 1974) Templin, R., "Aerodynamic Performance Theory for the NRC Vertical Axis Wind Turbine," National Research Council Canada Report LTR-LA-160, 1974.
- (Veers 1984) Veers , P., "Modeling Stochastic Wind Loads on Vertical Axis Wind Turbines," Sandia National Labs Report SAND83-1909, 1984.
- (Walters 1975) Walters, R., "Innovative Wind Machines," West Virginia University, 1975.
- (Wetzel 2009) Wetzel, D., Griffin, J., Liu, F., Cattafesta, L., "An Experimental Study of Circulation Control on an Elliptic Airfoil," AIAA Technical Paper Series, Paper No. 2009-4280, 2007.
- (Williams 1969) Williams, R. M. "Some Research on Rotor Circulation Control," CAL/AVLABS Symposium: Aerodynamics of Rotary Wing and V/STOL Aircraft. Buffalo, New York, June 1969.
- (Wilson 1974) Wilson , R., Lissaman, P., "Applied Aerodynamics of Wind Powered Machines," Technical Report NSF-RA-N-74-113, Oregon State University, 1974.
- (Wood 1981) Wood, N. J., "The Aerodynamics of Circulation Control Aerofoils," Stanford University Technical Report (SU-JIAA TR-41). Stanford, CA, 1981.
- (Wolfe 1981) Wolfe, W., *Analysis of Test Results for the WVU Straight Bladed Darrieus Wind Turbine*, Thesis, West Virginia University, Morgantown, West Virginia, 1978.

Appendix A. Source Code Listing

The MATLAB code of the 2D and 3D vortex models, double multiple stream tube model, the airfoil controller, and the solidity matching scheme are presented. The code is divided into the following files:

<u>File Name</u>	<u>Description</u>
VortexModelComplex	2D Vortex Model
CCVAWTController	2D vortex and momentum model airfoil controller
VortexModel3D	3D vortex model
MomentumModel	Double multiple stream tube momentum model
CCVAWT3DController	3D vortex airfoil controller
FindCuCtNeededToMatch	Solidity matching scheme which finds the blowing coefficient
FindPowerForCCEllipse	Deteremins the power produced by a VAWT
FindSolAndWRForEllipse	Searches for the best solidity and wR
VelBLayer	Calculates the vortex scale
VelFillament	Determines the 3D induced velocity

A.1 VortexModelComplex

```
%Copyright (c) Jay P. Wilhelm 2010, All Right Reserved

%VortexModelComplex class
%Purpose: Class which is the 2D Vortex Model

classdef VortexModelComplex < hgsetget

    properties
        %constants for the sim
        dtheta
        TSR
        C_R
        Nb
        TimeStepsPerRev

        %status of the vawt
        bladePos
        bladeG
        bladeRelVel
        obladePos
        obladeG
        vorPos
        vorG

        %settings
        bRemoveFarPoints
        DeleteThreshold
        mController;
        Use_Cuda
    end
end
```

```

        Use_Multiple_CPU
    end

    methods
        function vm = set.Use_Cuda(vm, val)
            vm.Use_Cuda = val;
        end
        function vm = set.Use_Multiple_CPU(vm, val)
            vm.Use_Multiple_CPU = val;
        end

        %the constructor function
        function vm=VortexModelComplex(iTimeStepsPerRev,...
            iNb,iTSR,iC_o_R,iController)
            vm.mController=iController;
            vm.Use_Cuda=1;
            vm.TimeStepsPerRev=iTimeStepsPerRev;
            vm.dtheta= (2*pi)/iTimeStepsPerRev;
            vm.TSR=iTSR;
            vm.C_R=iC_o_R;
            vm.Nb=iNb;
            vm.Init();
            vm.bRemoveFarPoints=0;
            vm.DeleteThreshold=6;
        end

        %called to init the turbine state
        function Init(vm)
            for ii=0:vm.Nb-1
                vm.bladePos(ii+1)=-exp(1i*(2*pi*ii)/(vm.Nb));
            end
            vm.bladeRelVel=-1*ones(1,vm.Nb)-1i.*vm.TSR.*vm.bladePos;
            alpha_list=angle(1i.*conj(vm.bladePos).*vm.bladeRelVel);
            [cld_list,cu,re]=vm.FindCl(alpha_list,vm.bladePos);
            vm.bladeG=-0.5.*vm.C_R.*real(cld_list).*abs(vm.bladeRelVel);
            vm.vorPos=0;
            vm.vorG=0;
        end

        %run range of TSR
        function [md,Cp]=RunRange(vm,tsr,maxi,debug)
            for ii=1:length(tsr)
                vmz(ii)=vm;
            end
            %run in parallel
            parfor ii=1:length(tsr)
                vmz(ii).TSR=tsr(ii);
                vmz(ii).Init();
                [md(ii)]=vmz(ii).RunUntilConverged(maxi,debug);
                Cp(ii)=md(ii).mcp;
            end
        end

        %run revolutions until convergence is met
        function [md] = RunUntilConverged(vm,maxi,debug)
            tic;
            %only rotate maxi times
            for ii=1:maxi
                %Do one complete revolution
                [md]=vm.RunOneRev();
                md.iters=ii;
                mcp=md.mcp;

                if(ii>4)
                    err = abs(mcp-old_mcp);
                    if(debug)
                        fprintf('TSR %g run:%g cp:%g err:%g\n',...
                            vm.TSR,ii,mcp,err);
                    end
                end
            end
            %convergence criteria

```

```

        if(err <= 0.01)
            md.IterationTime=toc;
            iters=ii;
            return;
        end

        %negative Cp
        if(mcp<0)
            md.IterationTime=toc;
            iters=ii;
            return;
        end
    else
        if(debug)
            fprintf('TSR %g run:%g cp:%g \n',vm.TSR,ii,mcp);
        end
    end
    md.IterationTime=toc;
    iters=ii;
    old_mcp=mcp;
end
end

%do one revolution
%save the result
function [md]=RunOneRev(vm)
    %place to save the result
    md=ModelDataHolder();

    for ii=1:vm.TimeStepsPerRev
        [theta(ii,:) Ft(ii,:) Vrel(ii,:) alpha(ii,:) Cp(ii,:) ...
         CpBlade(ii,:) cl(ii,:) ct(ii,:) cu(ii,:) ...
         Re_list(ii,:)]=vm.RunOneStep();
        md.theta=theta;
        md.Ft=Ft;
        md.CpBlade=CpBlade;
        md.cld=cl;
        md.cnt=ct;
        md.cu=cu;
        md.Re=Re_list;
        md.mcp=mean(Cp);
        md.Vrel=Vrel;
        md.alpha=alpha;
        md.vm=vm;
    end
end

%do one turbine step
%add to the vortex list
%determine performance
%save the state of the turbine
%works for N number of blades
function [theta Ft Vrel alpha Cp CpBlade ...
         cld_list cnt_list cu_list Re_list]=RunOneStep(vm)
    %save the old gamma
    vm.obladeG = vm.bladeG;

    %determine new blade position by
    %rotating one dtheta
    vm.bladePos = exp(1i.*vm.dtheta).*vm.bladePos;
    %determine the blade's rotational angle
    theta = angle(vm.bladePos);
    %save the vorticies & gamma in list
    biglist = [vm.bladePos vm.vorPos];
    biggamma = [vm.bladeG vm.vorG];

    %determine the induced velocity at the blade
    %include -1i influence from the wind
    vm.bladeRelVel=vm.VelFlow(vm.bladePos,biglist,biggamma)...

```



```

        -1i.*vm.TSR.*vm.bladePos;
Vrel = vm.bladeRelVel;

%find alpha
alpha_list = angle(1i.*conj(vm.bladePos).*vm.bladeRelVel);
alpha=alpha_list;
%lookup the CL, CD
[cld_list,cu_list,Re_list]=vm.FindCl(alpha_list,vm.bladePos);
vm.bladeG=-0.5.*vm.C_R.*real(cld_list).*abs(vm.bladeRelVel);
cnt_list=vm.FindCt(alpha_list,cld_list);
%determine the tangential coeff.
Ft=cnt_list.*real(vm.bladeRelVel.*conj(vm.bladeRelVel));
%determine the Cp
CpBlade=0.5.*vm.C_R.*vm.TSR.*imag(Ft);
Cp=0.5.*vm.C_R.*vm.TSR.*sum(imag(Ft));

%combine the shed vorticity
biggamma = [vm.bladeG vm.vorG];

%add the convected vortex positions to the list
if isempty(vm.vorPos)
    vm.vorPos = vm.bladePos;
    vm.vorG = vm.obladeG-vm.bladeG;
else
    vm.vorPos = vm.vorPos + vm.VelFlow(vm.vorPos,...
        biglist,biggamma).*vm.dtheta./vm.TSR;
    vm.vorPos = [vm.vorPos vm.bladePos];
    vm.vorG = [vm.vorG vm.obladeG-vm.bladeG];
end

%trim off far away points
%not used
if (vm.bRemoveFarPoints)
    newcount=1;
    for ii=1:length(vm.vorPos)
        if(abs(real(vm.vorPos(ii))) < vm.DeleteThreshold)
            newPoints(newcount)=vm.vorPos(ii);
            newG(newcount)=vm.vorG(ii);
            newcount=newcount+1;
        end
    end
    vm.vorPos=newPoints;
    vm.vorG=newG;
end
%pickup any errors
if(isnan(Cp))
    fprintf('theta %g alpha %g \r',...
        rad2deg(theta),rad2deg(alpha));
    error('NAN CP');
end
end

%induced velocity function
function vec=VelFlow(vm,p,vp,gamma)
%use CUDA
if(length(p)>5 && vm.Use_Cuda==1)
    vec=VelFlowCUDA(p,vp,gamma,vm.dtheta);
%stick to the CPU
else
    vec=zeros(1,length(p));
    for jj=1:length(p)
        for kk=1:length(vp)
            dp=p(jj)-vp(kk);
            vec(jj)=vec(jj)+1i*gamma(kk)*dp./...
                (2*pi*dp.*conj(dp)+(vm.dtheta^2./(2*pi)));
        end
    end
    vec=-1+vec;
end
end
end

```

```

    %%% Find the Cl
    %%% takes a list of alpha values
    function [cld,Cu,Re]=FindCl (vm,alpha_rad,theta)
        [cl cd Re Cu]=vm.mController.FindCl (alpha_rad,theta,vm);
        cld=cl+cd*1i;
    end

    %%% Find the Ct
    function cnt=FindCt (vm,alpha_rad,cld)
        cnt=conj (cld) .*exp (1i.*alpha_rad);
    end
end
end
end

```

A.2 VortexModel3D

```

%Copyright (c) Jay P. Wilhelm 2010, All Right Reserved
%CCVAWTController class
%Purpose: Class which handles all of the airfoil data
%         for the vortex 2D modeling of a VAWT

classdef VortexModel3D < hgsetget

    properties
        DLLFNAME;
        mController;

        %new for 3d
        RotorAspectRatio;
        Blades
        ElementsPerBlade
        Solidity
        TipSpeedRatio;

        GroundBoundrayLayerDelta;
        ThetaStep;
        ThetaRotMatrix;
        CaptureAreaShapeFactor;
        BladeCount;
        BECount;
        BEReyNumVinf;

        BEBegin;           %blade begin
        BEEnd;             %blade end
        BESpan;            %blade span
        BEChord;           %blade chord
        BERadius;          %blade radius
    end
end

```

```

BEChordUnitVec;    %blade chord unit vec
BESpanUnitVec;     %blade span unit vec
BENormUnitVec;     %blade norm unit vec
BERelVel;          %blade rel vel
BEProRelVel;       %blade pro rel vel
BEAoA;             %blade aoa
BEGamma;
BEForce;
BECp;
VFBegin;
VFEnd;
VFGamma;
ViciciousCore;

useCUDA;

end
properties (SetAccess=private)
    use_blowing
end
methods
    function vm = set.useCUDA(vm, val)
        vm.useCUDA = val;
        if (val)
            vm.DLLFNAME='GPUvelFillamentSet';
        else
            vm.DLLFNAME='CPUvelFillamentSet';
        end
    end
    function vm=VortexModel3D()
    end
    function vm=Init (vm, irH, iepB, iNb, iTSR, iC_o_R, iController)
        %vm.useCUDA=1;
        %vm.DLLFNAME='GPUvelFillamentSet';
        vm.RotorAspectRatio = irH;
        rH=irH;
        vm.Blades=iNb;
        nB=iNb;
        vm.TipSpeedRatio=iTSR;
        TSR=iTSR;
        vm.ElementsPerBlade=iepB;
        epB=iepB;
        vm.Solidity=iC_o_R;
        C_o_R=iC_o_R;
        Re=300e3;
        if ~isa(iController, 'CCVAWT3DController')
            error('wrong controller type');
        end
        vm.mController=iController;
        gbDelta=0;
        BEReyNumVinf=139160*ones(1,nB*epB); %%%CHECK THIS!
        z=rH*(0:epB^-1:1-epB^-1);
        theta=2*pi*(0:nB^-1:1-nB^-1);

        for ii=1:length(theta)
            itheta=theta(ii);
            for jj=1:length(z)

```

```

        iz = z(jj);
        bb((ii-1)*length(z)+jj,:)=[cos(itheta),...
            sin(itheta),iz];
        be((ii-1)*length(z)+jj,:)=[cos(itheta),...
            sin(itheta),iz+rH/epB];
        cu((ii-1)*length(z)+jj,:)=[-sin(itheta),...
            cos(itheta),0];
    end
end

dTheta = (pi/2)*(rH/epB);%2*pi/24;
Delta = dTheta^2 / (2*pi);
rM = [cos(dTheta) -sin(dTheta) 0;...
        sin(dTheta)  cos(dTheta) 0;...
        0            0          1]';
aF = 1/rH;

bbt=bb.*[ones(length(bb),1) ones(length(bb),1) ...
        zeros(length(bb),1)];
bet=be.*[ones(length(be),1) ones(length(be),1) ...
        zeros(length(be),1)];
b_temp=0.5*(bbt+bet);
s_temp=be-bb;
for ii=1:length(bbt)
    br(ii)=sqrt(dot(b_temp(ii,:),b_temp(ii,:)));
    bs(ii)=sqrt(dot(s_temp(ii,:),s_temp(ii,:)));
    su(ii,:)=normr(s_temp(ii,:));
end
br';
bs';
% su
%
nu=cross(su,cu);
nb=nB*epB;
b_del_alpha=zeros(1,nb);

bc=C_o_R*ones(1,nb);

bRe = Re*ones(1,nb);

bm=0.5*(bb+be);

gbDelta=0;
bvm = [TSR*bm(:,2)-velBLayer(abs(bm(:,3)),gbDelta)',...
        -TSR*bm(:,1),zeros(1,length(bm))']';
for ii=1:length(bvm)
    bvmt=bvm(ii,:);
    sut=su(ii,:);

    bvt=bvmt-(dot(bvmt,sut))*sut;
    bv(ii,:)=bvt;

    cut=cu(ii,:);
    nut=nu(ii,:);

```

```

        balpha(ii,:)=-atan2(dot(bvt,nut),-dot(bvt,cut));

        vs(ii)=dot(bvt,bvt);

end
%bv
%balpha
vs=vs;
v=sqrt(vs);
[cl cd]=vm.mController.FindCl(balpha,...
    zeros(1,length(balpha)),v);
cld=(cl+cd*1i);
cnt=conj(cld).*exp(1i.*balpha');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

bGamma = -0.5*bc.*bs.*real(cld).*v;

bF = 0.5*cnt.*bs.*bc.*vs;

bCp = TSR.*br.*imag(bF);

mkrange = 0:2*pi/3:2*pi*(1-1/3);
for ii=1:length(mkrange)
    mkt=mkrange(ii);
    mk(ii,:)=bb(1,:)+0.05.*[0,cos(mkt),sin(mkt)];
end
vb=[0 0 0];
ve=[0 0 0];
vGamma=0;

vm.TipSpeedRatio = TSR;
vm.GroundBoundrayLayerDelta=gbDelta;
vm.ThetaStep = dTheta;
vm.ThetaRotMatrix=rM;
vm.CaptureAreaShapeFactor = aF;
vm.BladeCount = nB;
vm.BECount = nb;
vm.BEReyNumVinf = BEReyNumVinf;

vm.BEBegin=bb;           %blade begin
vm.BEEnd=be;             %blade end
vm.BESpan=bs;            %blade span
vm.BEChord=bc;           %blade chord
vm.BERadius=br;          %blade radius
vm.BEChordUnitVec=cu;    %blade chord unit vec
vm.BESpanUnitVec=su;     %blade span unit vec
vm.BENormUnitVec=nu;     %blade norm unit vec
vm.BERelVel=vs;          %blade rel vel
vm.BEProRelVel=bv;       %blade pro rel vel
vm.BEAoA=balpha;         %blade aoa
vm.BEGamma=bGamma;
vm.BEForce=bF;
vm.BECp=bCp;
vm.VFBegin=[0 0 0];
vm.VFEnd=[0 0 0];

```

```

        vm.VFGamma=0;
        vm.ViciciousCore = Delta;

    end

function Cp=RunRange (vm,tsr,maxi,debug)
    for ii=1:length(tsr)
        vm.TSR=tsr(ii);
        vm.Init();
        [theta Ft Vrel alpha mcp iters clout ctout]=...
            vm.RunUntilConverged(maxi,debug);
        Cp(ii)=mcp;
    end
end

function [mcp, vmrun_all] = RunUntilConverged(vm,maxi,debug)
    for ii=1:maxi
        vmrun=RunOneRev (vm);

        if(ii==1)
            vmrun_all=vmrun;
        else
            vmrun_all=[vmrun_all;vmrun];
        end
        for zz=1:length(vmrun)
            tcp(zz)=sum(vmrun{zz}.BECp);
        end
        mcp=mean(tcp);

        TSR=vm.TipSpeedRatio;
        if(ii>4)
            err = abs(mcp-old_mcp);
            if(debug)
                fprintf('TSR %g run:%g cp:%g err:%g\n',...
                    TSR,ii,mcp,err);
            end
            if(err <= 0.01)
                iters=ii;
                return;
            end
        else
            if(debug)
                fprintf('TSR %g run:%g cp:%g \n',TSR,ii,mcp);
            end
        end
        iters=ii;
        old_mcp=mcp;
    end
end
function newObj=copyobj(thisObj)
    % Construct a new object based on a deep copy of the
current
    % object of this class by copying properties over.

```

```

newObj = feval(class(thisObj));

props = properties(thisObj);
for i = 1:length(props)
    newObj.(props{i}) = thisObj.(props{i});
end
end

function [vmrun]=RunOneRev (vm)

    revs=1;
    for ii=1:(2*pi*revs)/vm.ThetaStep
        vm.RunOneStep();
        vmrun{ii}=vm.copyobj();
    end

end

function vm=RunOneStep (vm)
    dTheta = vm.ThetaStep;
    rM      = vm.ThetaRotMatrix;
    TSR     = vm.TipSpeedRatio;
    gbDelta = vm.GroundBoundrayLayerDelta;
    aF      = vm.CaptureAreaShapeFactor;
    nB      = vm.BladeCount;
    bRe     = vm.BEReyNumVinf;
    bs      = vm.BESpan;
    bc      = vm.BEChord;
    br      = vm.BERadius;
    nb      = vm.BECount;
    bGamma  = vm.BEGamma;
    obb     = vm.BEBegin;
    obe     = vm.BEEnd;
    ocu     = vm.BEChordUnitVec;
    osu     = vm.BESpanUnitVec;
    onu     = vm.BENormUnitVec;
    obGamma = vm.BEGamma;
    ovb     = vm.VFBegin;
    ove     = vm.VFEnd;
    ovGamma = vm.VFGamma;
    Delta   = vm.VicicousCore;

    %rotate geometry, points & unit vectors
    bb=obb*rM;
    be=obe*rM;
    cu=ocu*rM;
    su=osu*rM;
    nu=onu*rM;

    %combine blade element and free vortex fillaments
    ab=[bb ; be ; bb ; obb; ovb];
    ae=[be ; obe; obb; obe; ove];

    bbe=0.5*(bb+be);

```

```

bvbeINF=[TSR*bbe(:,2)-velBLayer(abs(bbe(:,3)),gbDelta)',...
        -TSR*bbe(:,1),...
        zeros(1,length(bbe))'];

const_BoundFillamentIterations=3;
for ii=1:const_BoundFillamentIterations
    aGamma=[bGamma, bGamma, -bGamma, obGamma-bGamma, ovGamma];
    bvbe=bvbeINF+vm.velFillament(bbe,ab,ae,aGamma,Delta);
    for jj=1:length(bvbe)
        bv(jj,:)=bvbe(jj,:)-
dot(bvbe(jj,:),su(jj,:)).*su(jj,:);
    end

    for jj=1:length(bv)
        bvt=bv(jj,:);
        cut=cu(jj,:);
        nut=nu(jj,:);
        balpha(jj,:)=-atan2(dot(bvt,nut),-dot(bvt,cut));

        bvs(:,jj)=dot(bvt,bvt);
        bvm(:,jj)=sqrt(bvs(:,jj));
    end

    %
    [cl cd]=vm.mController.FindCl(balpha,...
        zeros(1,length(balpha)),bvm);
    cld=(cl+cd*1i);
    cnt=conj(cld).*exp(1i.*balpha)';
    %
    %no fillament scaling
    bGamma = -0.5*bc.*real(cld).*bvm;
    %linear fillament scaling
    %bGamma = -0.5*bc.*bs.*real(cld.*bvm);
    %sqrt fillament scaling
    %bGamma = -0.5*bc.*sqrt(bs).*real(cld.*bvm);
end

vb=[be ;bb ;obb];
ve=[obe ;obb ;obe];
vGamma=[bGamma' ; -bGamma' ; obGamma' -bGamma'];

%remove duplicates
new_i=1;
clear nvb nve nvb_gamma nve_gamma;
for ii=1:length(vb)
    vb_clean=1;
    ngamma=vGamma(ii);
    for jj=1:length(vb)
        if(mean(vb(ii,:)==vb(jj,:))==1 && ii~=jj)
            ngamma=sum([ngamma vGamma(jj)]);
        end
    end
    for jj=1:ii-1
        if(mean(vb(ii,:)==vb(jj,:))==1 && ii~=jj)

```



```

        vb_clean=0;
    end
end
if (vb_clean==1)
    nvb(new_i,:)=vb(ii,:);
    nvGamma(new_i)=ngamma;
    nve(new_i,:)=ve(ii,:);
    new_i=new_i+1;
end
end

vb=nvb;
ve=nve;
vGamma=nvGamma;

% Convect previously shed vortex fillaments
if (length(ovb(:,1)) > nb)
    ovncb=ovb(1:nb+nB,:);
    ovccb=ovb(nb+nB+1:length(ovb),:);
    ovccbet = [ovccb ; ove];
    mid=[-velBLayer(abs(ovccbet),gbDelta);...

zeros(1,length(ovccbet));zeros(1,length(ovccbet))]' ;
    ovccbet = ovccbet + (mid ...
        +
vm.velFillament(ovccbet,ab,ae,aGamma,Delta)) ...
        *dTheta/TSR;

    ovb = [ovncb;ovccbet(1:length(ovccb),:)] ;
    ove = ovccbet(length(ovccb)+1:(length(ovccb)+...
        length(ove)),:);

end
%
[c1 cd]=vm.mController.FindCl(balpha,...
    zeros(1,length(balpha)),bvm);
cld=(c1+cd*li);
cnt=conj(cld).*exp(li.*balpha');
%
bF = 0.5.*cnt.*bs.*bc.*bvs;
bCp = aF.*TSR.*br.*imag(bF);

vm.BEBegin=bb;           %blade begin
vm.BEEnd=be;             %blade end
vm.BEChordUnitVec=cu;    %blade chord unit vec
vm.BESpanUnitVec=su;     %blade span unit vec
vm.BENormUnitVec=nu;     %blade norm unit vec
vm.BERelVel=bvbe;        %blade rel vel
vm.BEProRelVel=bv;       %blade pro rel vel
vm.BEAoA=balpha;         %blade aoa
vm.BEGamma=bGamma;
vm.VFBegin=[vb;ovb];
vm.VFEnd=[ve;ove];
vm.VFGamma=[vGamma,ovGamma];
vm.BEForce=bF;
vm.BECp=bCp;

```

```

end

    %%% Find the Cl
    %%% takes a list of alpha values
    %%% need to include Re, Cu
    function [cld]=FindCl(vm,alpha_rad,theta)
        [cl
cd]=vm.mController.FindCl(alpha_rad,theta,vm.bladeRelVel);
        cld=cl+cd*1i;
    end
    function cnt=FindCt(vm,alpha_rad,cld)
        cnt=conj(cld).*exp(1i.*alpha_rad);
    end
    function velF=velFillament(vm,p3,p1,p2,Gam,delt)
        if(~vm.useCUDA)
            velF=velFillament(p3,p1,p2,Gam,delt);
        else
            vm.useCUDA
            N1=length(p1(:,1));
            N3=length(p3(:,1));
            vx = libpointer('singlePtr',zeros(1,N3));
            vy = libpointer('singlePtr',zeros(1,N3));
            vz = libpointer('singlePtr',zeros(1,N3));
            [vx,vy,vz]=VelFillament3D(p1(:,1),p1(:,2),p1(:,3),...
                p2(:,1),p2(:,2),p2(:,3),...
                p3(:,1),p3(:,2),p3(:,3),...
                Gam,delt,vm.useCUDA);
            velF=[vx ; vy ; vz]';
        end
    end
end
end
end

```

A.3 CCVAWTController

```

%Copyright (c) Jay P. Wilhelm 2010, All Right Reserved

%CCVAWTController class
%Purpose: Class which handles all of the airfoil data
%         for the vortex 2D modeling of a VAWT

classdef CCVAWTController < hgsetget

```

```

properties
    %Basic properties of the 2d controller
    UseNACA0018Only
    MatchToVM
    UseCIRAEllipse
    EllipseAoA
    EllipseCL
    EllipseCD
    EllipseCu
    EllispeCT
    NACA
    UseConstantBlowingRate;
    UseConstantBlowingCoeff;
    UsePartitionedBlowing
    NumberOfPartitions
    PartitionsState
    AoABase

    PartitionLocations

    axisAlpha
    axisRe
    tableCl
    tableCd

    UseRe
    ReVinf

    %holds the airfoil data
    UseKwon2005
    KwonAoA
    KwonCL
    KwonCD
    tableCD10Blowing
    tableCL10Blowing
    tableCDNoBlowing
    tableCLNoBlowing
    tableBlowingCL
    tableBlowingCD
    tableAoA
    tableAoACL
    tableAoACD
    tableRe
    tableCu
    tableCu_max
    tableCu_min

    cuLoc=3;
    ReLoc=2;
    aoaLoc=1;
    blowLevel;
    BladeChord_inches
    JetSlotHeight_inches
    Vinf_fts

```

```

Beta
end
methods
    %Called to create the class
    function obj = CCVAWTController()
        %initial states of the class
        obj.UseCIRAEllipse=0;
        obj.UseDeltaFactors=0;
        obj.UseNACA0018Only=0;
        obj.UseConstantBlowingRate=0;
        obj.UseConstantBlowingCoeff=0;
        obj.UsePartitionedBlowing=0;
        obj.NumberOfPartitions=0;
        obj.PartitionsState=0;
        obj.AoABase=0;
        obj.UseRe=1;
        obj.blowLevel=0;

        %Load the NACA0018 airfoil data
        load('naca0018_re_fixed');
        obj.tableCl=naca0018_cl_allfixed;
        obj.tableCd=naca0018_cd_allfixed;
        obj.axisAlpha=naca0018alphaaxisfixed;
        obj.axisRe=naca0018reaxis;

        if(max(obj.axisAlpha) > 100) %force rad conversion
            obj.axisAlpha=deg2rad(obj.axisAlpha);
        end

        %Load the CC-Ellipse airfoil data
        load('ellipse_chad_3_full.mat');
        obj.EllipseAoA=deg2rad(eaoa);

        eaoa=eaoa';
        ecu=ecu./100;
        xover = find(eaoa>0);
        xover2 = find(eaoa>=0);

        %correct for slot switch
        ecl1=[flipud(-ecl(xover,:)); ecl(xover2,:)];
        ecd1=[flipud(ecd(xover,:)); ecd(xover2,:)];

        xover = find(eaoa<0);
        xover2 = find(eaoa<=0);

        lecu=length(ecu);
        ecl2=[(ecl(xover,2:lecu)); -flipud(ecl(xover2,2:lecu))];
        %correct for slot switch
        ecl=[fliplr(ecl2) ecl1];
        ecd=[ecd1(:,2:lecu) ecd1];
        ecu = [fliplr(-ecu(2:length(ecu))) ecu];

        obj.EllipseCL=ecl';
        obj.EllipseCD=ecd';
        obj.EllipseCu=ecu;
    end
end

```

```

        %pre calc the Ct
        for ii=1:length(ecu)
            ect(:,ii) =          ecl(:,ii).*sind(eaoa)
ecd(:,ii).*cosd(eaoa);
        end
        obj.EllispeCT=ect;
    end
    %set the blowing level
    function LoadBlowing(obj,blowLevel)
        obj.blowLevel=blowLevel;
    end
    %set the solidity to match
    function obj=set.MatchToVM(obj,val)
        obj.MatchToVM=val;
    end
    %set to use NACA0018 only
    function obj = set.UseNACA0018Only(obj,val)
        obj.UseNACA0018Only = val;
    end
    function obj = set.JetSlotHeight_inches(obj,val)
        obj.JetSlotHeight_inches = val;
    end
    %use partitioned blowing scheme
    function obj = set.UsePartitionedBlowing(obj,val)
        obj.UsePartitionedBlowing = val;
    end
    function obj = set.NumberOfPartitions(obj,val)
        obj.NumberOfPartitions = val;
        obj.PartitionLocations=(0:val-1)*360/val;
        fixhigh=find(obj.PartitionLocations>=360);
        obj.PartitionLocations(fixhigh)= ...
            obj.PartitionLocations(fixhigh) - 360;
    end
    function obj = set.UseCIRAEllipse(obj,val)
        obj.UseCIRAEllipse=val;
    end
    %set partitions state to blowing or non, must be vector
    function obj = set.PartitionsState(obj,val)
        obj.PartitionsState = val;
        obj.set('NumberOfPartitions',length(val));
    end
    %set to use constant blowing coeff scheme
    function obj = set.UseConstantBlowingCoeff(obj,val)
        obj.UseConstantBlowingCoeff=val;
    end;
    %set to use constant blowing rate scheme
    function obj = set.UseConstantBlowingRate(obj,val)
        obj.UseConstantBlowingRate=val;
    end;
    %set the beta parameter of blowing
    function obj = set.Beta(obj,val)
        obj.Beta = val;
    end
    function obj = set.AoABase(obj,val)%radians
        obj.AoABase = val;

```

```

end
function obj = set.UseRe(obj,val)
    obj.UseRe = val;
    if(val)
        obj.UpdateReVinf();
    end
end
function obj = set.BladeChord_inches(obj,val)%meters
    obj.BladeChord_inches = val;
    obj.UpdateReVinf();
end
function obj = set.Vinf_fts(obj,val)%meters/second
    obj.Vinf_fts = val;
    obj.UpdateReVinf();
end
function UpdateReVinf(obj)%meters
    Vinf_ms=obj.Vinf_fts*0.3048;
    obj.ReVinf=Vinf;
end
%the main findcl, gets called from the VM
function [cl cd Re Cu Cu_t]=FindCl(obj,alpha_rad,ibladepos,vm)
    vrel=vm.bladeRelVel;
    Cu=zeros(1,length(alpha_rad));
    PartitionLocations=...
        [obj.PartitionLocations obj.PartitionLocations(1)+360];
    %init for speed
    cl=zeros(1,length(alpha_rad));
    cd=zeros(1,length(alpha_rad));
    Re=zeros(1,length(alpha_rad));
    %loop through each blade by its position
    for jj=1:length(ibladepos)
        bladepos=ibladepos(jj);
        theta=rad2deg(angle(bladepos));
        %fix matlab's error with == 0
        if(theta <= 0.0001 && theta >= -0.0001)
            theta=0;
        end
        %make sure its within -180
        if(theta >= -180 && theta < 0)
            theta = 360+theta;
        end
        FOUND=0;
        for ii=1:obj.NumberOfPartitions
            %quick case for no or 1 partition
            if(obj.PartitionsState(ii)==0)
                Cu(jj)=0.0;
            end
            if(obj.NumberOfPartitions==1)
                [cl(jj) cd(jj) Re(jj) Cu(jj)]=...
                    obj.FindClFromTable(alpha_rad(jj),...
                        vrel(jj),Cu(jj),bladepos,vm);
                FOUND=1;
                break;
            end
        end

        loc = PartitionLocations(ii);

```

```

        loc_next = PartitionLocations(ii+1);
        if(theta >= loc && theta < loc_next)
            [cl(jj) cd(jj) Re(jj) Cu(jj)]=...
                obj.FindClFromTable(alpha_rad(jj),...
                    vrel(jj),Cu(jj),bladeapos,vm);
            FOUND=1;
            break;
        end
    end
end
if(~FOUND)
    fprintf('Blade:%f AoA:%f \r\n',theta,...
        rad2deg(alpha_rad(jj)));
    error('Could not find CL CD ???!?!');
end

end

end
%uses the particular blowing scheme as defied in the class
function [cl cd Re Cu_o]=FindClFromTable(obj,...
    alpha_rad,vrel,cu,bladeapos,vm)
    if(obj.UseCIRAellipse==1 && obj.UseDeltaFactors==0 ...
        && obj.UseConstantBlowingCoeff==0 &&...
        obj.UseConstantBlowingRate==0)
        cu=0;
        %do this for solidity matching
        if(isobject(obj.MatchToVM))
            TSR=vm.TSR;
            C_R=vm.C_R;
            AoA=alpha_rad;
            Vrel=vm.bladeRelVel(1);
            eaoa=obj.EllipseAoA;
            ect=obj.EllipseCT;
            ecu=obj.EllipseCu;

            %issue with matlab's digit accuracy
            dumbfix=1000;
            theta_at=fix(angle(bladeapos)*dumbfix)/dumbfix;
            fit_theta=fix(obj.MatchToVM.theta(:,1)*dumbfix)...
                /dumbfix;
            theta_go=find(fit_theta>=theta_at,1,'First');
            CpSol2=obj.MatchToVM.CpBlade(theta_go,1);
            %lookup the Cu
            [CuN          CtP          CtN          Cp_p]          =
FindCuCtNeededToMatch(CpSol2,...
                TSR,C_R,Vrel,AoA,eaoa,ect,ecu);
            cu=CuN;
        end
        [cl cd Re Cu_o]=FindEllipseBlowingCl(obj,...
            alpha_rad,vrel,cu,bladeapos);
    elseif(obj.UseNACA0018Only==1 && obj.UseDeltaFactors==0
&&...
        obj.UseConstantBlowingRate==0)
        [cl cd Re]=obj.FindNACA0018Cl(alpha_rad,vrel);Cu_o=0;
    elseif(obj.UseConstantBlowingCoeff)
        cu=obj.blowLevel/100;
        [cl cd Re Cu_o]=FindEllipseBlowingCl(obj,alpha_rad,...

```

```

        vrel,cu,bladeapos);
    else
        obj.UseNACA0018Only
        error('Unknown Config');
    end
end
%find the CCellipse c1 & cd with blowing
function [c1 cd Re Cu_o]=FindEllipseBlowingCl(obj,alpha_rad,...
    vrel,cu,bladeapos)
    c1 = interp2(obj.EllipseAoA,obj.EllipseCu,obj.EllipseCL,...
        alpha_rad,cu);
    cd = interp2(obj.EllipseAoA,obj.EllipseCu,obj.EllipseCD,...
        alpha_rad,cu);

    if(isnan(c1) || isnan(cd))
        fprintf('alpha %g \r',rad2deg(alpha_rad));
        error('NAN CP');
    end
    Re=300e3;
    Cu_o=cu;
end
%find the CCNACA0018 c1 & cd with blowing
function [c1 cd Re Cu_o]=FindBlowingCl(obj,alpha_rad,vrel,...
    cu,bladeapos)
    if(obj.UseNACAxxxxOnly)
        Cu_o=0.0;
        if(obj.UseNACA0018Only)
            [c1 cd Re]=FindNACA0018Cl(obj,alpha_rad,vrel);
        elseif(obj.UseNACA0012Only)
            c1 = interp1(obj.axisAlpha, obj.tableCl,alpha_rad);
            cd = interp1(obj.axisAlpha, obj.tableCd,alpha_rad);
            Re=40e3;
        end
        return;
    end
    Re=300e3;
    if(cu < 0 || obj.blowLevel==1)
        cu=0;
    elseif(cu > obj.tableCu_max)
        cu=obj.tableCu_max;
    end
    Cu_o=cu;
    alpha_rad2=alpha_rad;
    flip = 0;
    if(alpha_rad < 0)
        flip=1;
    end
    if(flip)
        alpha_rad2=-alpha_rad;
    end
    %fixed data format
    c1 = interp2(obj.tableAoACL,obj.tableCu,...
        obj.tableBlowingCL,alpha_rad2,cu);
    cd = interp2(obj.tableAoACD,obj.tableCu,...
        obj.tableBlowingCD,alpha_rad2,cu);

```



```

        if(flip)
            cl=-cl;
        end
        %ALL CU, ALL RE
        %cl = interp2(obj.tableCu,obj.tableRe,obj.tableAoA,...
        %obj.tableBlowingCL(:, :, :), cu,Re,alpha_rad);
        %cd = interp2(obj.tableCu,obj.tableRe,obj.tableAoA,...
        %obj.tableBlowingCD(:, :, :), cu,Re,alpha_rad);
        %ALL CU, ONE RE
        %cl = interp2(obj.tableAoA,obj.tableCu, ...
        %squeeze(obj.tableBlowingCL(:,1,:)), alpha_rad,cu);
        %cd = interp2(obj.tableAoA,obj.tableCu, ...
        %squeeze(obj.tableBlowingCD(:,1,:)), alpha_rad,cu);
        %ONE CU, ALL RE
        %cl = -1.*interp2(obj.tableAoA,obj.tableRe, ...
        %squeeze(obj.tableBlowingCL(CU_i, :, :)), alpha_rad,Re);
        %cd = -1.*interp2(obj.tableAoA,obj.tableRe, ...
        %squeeze(obj.tableBlowingCD(CU_i, :, :)), alpha_rad,Re);
        %ONE CU, ONE RE
        %cl = interp1(obj.tableAoA,squeeze(...
        %obj.tableBlowingCL(1,1,:)), alpha_rad2);
        %cd = interp1(obj.tableAoA,squeeze(...
        %obj.tableBlowingCD(1,1,:)), alpha_rad2);
        %error check
        if(sum(isnan(cl)) || sum(isnan(cd)))
            fprintf('alpha %g \r',rad2deg(alpha_rad));
            error('NAN CL or CD');
        end
    end
end
%Kwon has ellipse with > 2pi slope
function [cl cd Re]=FindKwon2005Cl(obj,alpha_rad,vrel)
    Re=300e3;
    if(alpha_rad > max(obj.KwonAoA) || ...
        alpha_rad < min(obj.KwonAoA))
        [cl cd Re]=obj.FindNACA0018Cl(alpha_rad,vrel);
    else
        cl = interp1(obj.KwonAoA, obj.KwonCL, alpha_rad);
        cd = interp1(obj.KwonAoA, obj.KwonCD, alpha_rad);
    end
end
%Just the sheldahl NACA0018
function [cl cd Re]=FindNACA0018Cl(obj,alpha_rad,vrel)
    alpha_rad=alpha_rad+obj.AoABase;
    Re=300e3;
    cl = interp2(obj.axisRe,obj.axisAlpha, obj.tableCl,...
        Re,alpha_rad);
    cd = interp2(obj.axisRe,obj.axisAlpha, obj.tableCd,...
        Re,alpha_rad);
end
%Computes the Ct from CL & CD in complex form
function cnt=FindCt(vm,alpha_rad,cld)
    cnt=conj(cld).*exp(1i.*alpha_rad);
end
end
end
end

```

A.4 MomentumModel

```
%Copyright (c) Jay P. Wilhelm 2010, All Right Reserved

%MomentumModel class
%Purpose: Class which is the momentum model

classdef MomentumModel < hgsetget
    %basic properties
    properties
        solve          %solidity
        mController
        dtheta
    end
    methods
        %construct the momentum model
        function mm=MomentumModel(isol,icontroller)
            mm.mController=icontroller;
            mm.sol=isol;
            mm.dtheta=deg2rad(1);
        end
        function mm=set.dtheta(mm,val)
            mm.dtheta=val;
        end
        function mm=set.sol(mm,val)
            mm.sol=val;
        end

        %Solve the Upwind VAWT, return Cp
        function Cp=SolveUpwindRange(mm,tsr)
            for ii=1:length(tsr)
                zcp=mm.SolveUpwind(tsr(ii));
                Cp(ii)=mean(zcp);
                fprintf('SOLVED U TSR:%g Cp:%g\n',tsr(ii),Cp(ii));
            end
        end

        %Solve the Upwind VAWT, return Cp,Ft,a,AoA,V
        function [Cp,Ftheta,a,alpha,Vrel] = SolveUpwind(mm,tsr)
            theta_rad = [-pi/2:mm.dtheta:pi/2];
            parfor ii=1:length(theta_rad)
                [Cp(ii) Ftheta(ii) a(ii) alpha(ii) Vrel(ii)]=...
                    mm.SolveUTube(tsr,theta_rad(ii));
            end
        end

        %Solve an Upwind Tube
        function [Cp,Ftheta,a,alpha,Vrel,Fdiff] = ...
            SolveUTube(mm,tsr,theta_rad)
```

```

%check to make sure we are upwind
if(theta_rad == pi/2 || theta_rad == -pi/2 ...
    || theta_rad == (3*pi)/2)
    a=0;
    Vrel=0;
    alpha=0;
    Fdiff=0;
else
    %try to solve for 'a'
    try
        [a FVal ExitFlag] = fzero(@(x) ...
            FindUTubeZero(mm,tsr, theta_rad,x),[-1 1]);
    catch
        fprintf('no a zero @ theta %g tsr %g sol %g\n',...
            rad2deg(theta_rad),tsr,mm.sol);
        a=0;
    end
    %use the calculated 'a' and get the velocity
    [Vrel, alpha,Fdiff,Faer,Fmom]=...
        mm.EvaluateUTube(tsr,theta_rad,a);
end

%use the calculated 'a' and get the forces
[Ftheta] = mm.GetUThetaForce(a,theta_rad,tsr);
%calc the Cp
Cp = 0.5*tsr*Ftheta;
end

%Solve the Upwind then Downwind VAWT, return Cp
function Cp=SolveUDRange(mm,tsr)
    for ii=1:length(tsr)
        zcp=mm.SolveUD(tsr(ii));
        Cp(ii)=mean(zcp);
        fprintf('SOLVED UD TSR:%g Cp:%g\n',tsr(ii),Cp(ii));
    end
end

%Solve the Upwind then Downwind VAWT, return Cp,Ft,a,AoA,V
function [Cp,Ftheta,a,alpha,Vrel] = SolveUD(mm,tsr)
    theta_rad = [-pi/2:mm.dtheta:(3*pi)/2];
    for ii=1:length(theta_rad)
        if(theta_rad(ii) <= pi/2)
            [Cp(ii) Ftheta(ii) a(ii) alpha(ii) Vrel(ii)]=...
                mm.SolveUTube(tsr,theta_rad(ii));
            iswitch=ii;
        else
            check=ii-2*(ii-ismatch);
            %need to feed the downwind the correct 'a'
            a_o=a(check);
            [Cp(ii) Ftheta(ii) b(ii) alpha(ii) Vrel(ii)]=...
                mm.SolvedTube(tsr,a_o,theta_rad(ii));
        end
    end
end

%Solve an Downwind Tube

```

```

%Requires 'a' from upwind
function [Cp,Ftheta,b,alpha,Vrel,Fdiff] = ...
    SolveDTube(mm,tsr,a,theta_rad)
    %check to make sure this is a downwind problem
    if(theta_rad == pi/2 || theta_rad == -pi/2 ...
        || theta_rad == (3*pi)/2)
        b=0;
        Vrel=0;
        alpha=0;
        Fdiff=0;
    else
        try
            %solve for b
            [b FVal ExitFlag] = ...
                fzero(@(x) FindDTubeZero(mm,tsr, ....
                    theta_rad,a,x),[-1 1]);
        catch
            fprintf('no b zero @ theta %g tsr %g sol %g\n',...
                rad2deg(theta_rad),tsr,mm.sol);
            b=-1;
        end
        %use the calculated 'b' and get the velocity
        [Vrel, alpha,Fdiff,Faer,Fmom]=...
            mm.EvaluatedTube(tsr,theta_rad,a,b);
    end
    %use the calculated 'b' and get the force
    [Ftheta] = mm.GetDThetaForce(a,b,theta_rad,tsr);
    %calculate Cp
    Cp = 0.5*tsr*Ftheta;

end

%use matlab search function to solve for a
function y = FindUTubeZero(mm,tsr, theta,a)

    [Ur, alpha_deg,Fdiff,Faer,Fmom]...
        = mm.EvaluateUTube(tsr, theta,a);

    y=Fdiff;
end

%use matlab search function to solve for b
function y = FindDTubeZero(mm,tsr, theta,a,b)

    [Ur, alpha_deg,Fdiff,Faer,Fmom]...
        = mm.EvaluatedTube(tsr, theta,a,b);

    y=Fdiff;
end

%Evaluate an Upwind tube
function [Vrel, alpha_deg,Fdiff,Faer,Fmom]=...
    EvaluateUTube(mm,tsr,theta_rad,a)
[Vrel, alpha_deg,Fdiff,Faer,Fmom]=...
    EvaluateTube(mm,tsr,theta_rad,a,0);

```

```

end
%Evaluate an Downwind tube
function [Vrel, alpha_deg, Fdiff, Faer, Fmom]=...
    EvaluateDTube(mm,tsr,theta_rad,a,b)
[Vrel, alpha_deg, Fdiff, Faer, Fmom]=...
    EvaluateTube(mm,tsr,theta_rad,a,b);
end
%Evaluate a tube
function [Vrel, alpha_deg, Fdiff, Faer, Fmom]=....
    EvaluateTube(mm,tsr,theta_rad,a,b)

    %determine the proper 'mid' parameter
    %ie, is this upwind or downwind
    theta_deg=rad2deg(theta_rad);
    if(theta_deg <= 90 && theta_deg >= -90)
        mid = (1-a);
    else
        mid = (1-b)*(1-2*a);
    end

    %calculate the velocity & AoA
    Vrel = sqrt(mid^2-2*tsr*mid*sin(theta_rad)+tsr^2);
    alpha_rad=mm.GetAlphaRad(theta_rad,tsr,a,b);
    alpha_deg=rad2deg(alpha_rad);
    %lookup the Cl & Cd
    [Cl Cd]=mm.mController.FindCl(alpha_rad,theta_rad,Vrel);

    %calc the momentum force
    Fmom = mm.GetMomForce(a,b,theta_rad);

    arad=alpha_rad;
    trad=theta_rad;
    %calc the aerodynamic force
    Faer=mm.GetAeroForce(mm.sol,Vrel,alpha_rad,theta_rad,Cl,Cd);
    %calc the difference(in positive)
    Fdiff = Fmom+Faer;
end

%Calc the upwind or downwind Theta direction force
function [Ftheta] = GetUThetaForce(mm,a,theta_rad,tsr)
[Ftheta] = GetThetaForce(mm,a,0,theta_rad,tsr);
end
function [Ftheta] = GetDThetaForce(mm,a,b,theta_rad,tsr)
[Ftheta] = GetThetaForce(mm,a,b,theta_rad,tsr);
end
%Calc a theta force
function [Ftheta] = GetThetaForce(mm,a,b,theta_rad,tsr)
    %determine if this is upwind or downwind
    theta_deg=rad2deg(theta_rad);
    if(theta_deg <= 90 && theta_deg >= -90)
        mid = (1-a);
    else
        mid = (1-b)*(1-2*a);

```

```

end
%calc the velocity
nvrel = sqrt(mid^2-2*tsr*mid*sin(theta_rad)+tsr^2);
%calc the AoA
alpha_rad=mm.GetAlphaRad(theta_rad,tsr,a,b);
%lookup the CL & CD
[Cl Cd]=mm.mController.FindCl(alpha_rad,theta_rad,nvrel);
%determine the theta force
Ftheta=mm.sol*nvrel^2*...
    (-Cd*cos(alpha_rad)...
    +Cl*sin(alpha_rad));
end

%calc the aero force
function Faer=GetAeroForce(mm,sol,Ur,theta_rad,alpha_rad,Cl,Cd)
    arad=alpha_rad;
    trad=theta_rad;
    %calculate the aerodynamic force
    Faer=sol*(Ur)^2*(Cl*cos(trad+arad)+Cd*sin(trad+arad));
end

%calculate the upwind or downwind mom. force
function [Fmom] = GetUMomForce(mm,a,theta_rad)
    [Fmom] = GetMomForce(mm,a,0,theta_rad);
end
function [Fmom] = GetDMomForce(mm,a,b,theta_rad)
    [Fmom] = GetMomForce(mm,a,b,theta_rad);
end

%calculate a mom. force
function [Fmom] = GetMomForce(mm,a,b,theta_rad)
    %use the upwind or downwind method
    %glauart disc formula
    theta_deg=rad2deg(theta_rad);
    if(theta_deg <= 90 && theta_deg >= -90)
        if a < 0.4
            Fmom = pi*cos(theta_rad)*4 * a * (1 - a);
        else
            Fmom = pi*cos(theta_rad)*(26 * a + 4) / 15;
        end
    else
        if b < 0.4
            Fmom = pi*cos(theta_rad)*4 * b * (1 - b)*(1-2*a)^2;
        else
            Fmom = (pi*cos(theta_rad)*26*(b + 4)*(1-2*a)^2) /
15;
        end
    end
end

%find the upwind or downwind AoA
function alpha_rad=GetUAlphaRad(mm,theta_rad,tsr,a)
    alpha_rad=GetAlphaRad(mm,theta_rad,tsr,a,0);
end
function alpha_rad=GetDAlphaRad(mm,theta_rad,tsr,a)
    alpha_rad=GetAlphaRad(mm,theta_rad,tsr,a,b);

```

```

end

%find the AoA
function alpha_rad=GetAlphaRad(mm,theta_rad,tsr,a,b)
    %use the upwind or downwind method
    theta_deg=rad2deg(theta_rad);
    if(theta_deg <= 90 && theta_deg >= -90)
        mid = (1-a);
    else
        mid = (1-b)*(1-2*a);
    end
    alpha_rad = atan2(-cos(theta_rad), (-
sin(theta_rad)+tsr/mid));
end
end
end

```

A.5 VelFillament

```

%Copyright (c) Jay P. Wilhelm 2010, All Right Reserved
%velFillament
%Purpose: Compute the fillaments velocity
%Inputs: p3, list of test points
%         p1, p2, list of vortex fillaments
%         Gam, gamma value of the vortex
%         delt, delta theta parameter
%Outputs: velF, induced velocity at p3

function velF=velFillament(p3,p1,p2,Gam,delt)
    if(length(p2) ~= length(p1) && length(p1) ~= length(Gam))
        error('wrong list lengths velFillament');
        velF=0;
        return;
    end
    for ii=1:length(p3(:,1))
        velF(ii,:)= [0 0 0];
        for jj=1:length(p1(:,1))
            a=p2(jj,:)-p1(jj,:);
            b=p2(jj,:)-p3(ii,:);
            c=p1(jj,:)-p3(ii,:);

            aa=dot(a,a);
            bb=dot(b,b);
            cc=dot(c,c);

            if(aa>0 && bb>0 && cc>0)
                c_x_a=cross(c,a);
                velF_left=(Gam(jj)*c_x_a)/(4*pi*dot(c_x_a,c_x_a)+delt);
                velF_right=(dot(a,b)/sqrt(bb))-(dot(a,c)/sqrt(cc));
                velF(ii,:)=velF(ii,:)+velF_left*velF_right;
            end
        end
    end
end

```

```

        end
    end
end

```

A.6 VelBLayer

```

%Copyright (c) Jay P. Wilhelm 2010, All Right Reserved

%velBLayer
%Purpose: Compute the bound velocity layer, 2 part function
%Inputs: distance, delta theta
%Outputs: velocity scale

function velB=velBLayer(iz,del)
for ii=1:length(iz)
    z=iz(ii);
    if(del==0)
        velB(ii)=ones(1,length(z));
    else
        x=z/del;
        if(x<=1)
            velB(ii)=x*(2-x);
        else
            velB(ii)=1;
        end
    end
end
end

```

A.7 ModelDataHolder

```

%Copyright (c) Jay P. Wilhelm 2010, All Right Reserved
%ModelDataHolder class
%Purpose: Class which holds the model data

classdef ModelDataHolder < hgsetget
    properties
        dtheta
        iters
        mcp
        theta
        alpha
        clid
        cnt
        cu
    end
end

```



```

        CpBlade
        BladePos
        BladeG
        vorPos
        vorG
        Ft
        Vrel
        Re

        IterationTime
        vm
    end
    methods
        function md=ModelDataHolder()
            md.iters=0;
        end
    end
end
end

```

FindCuCtNeededToMatch

```

%Copyright (c) Jay P. Wilhelm 2010, All Right Reserved

%FindCuCtNeededToMatch
%Purpose: Compute the Cu needed to match solidity
%Inputs: Coefficient of performance of solM
%         Tip Speed Ratio
%         Chord over Radius
%         Velocity relative (complex)
%         AoA degress
%         ellipse or airfoil AoA vector (degree)
%         ellipse or airfoil coef. tangent vector
%         ellipse or airfoil coef. blowing vector
%Outputs: Blowing Coeff. Needed
%         Coef. tangent possible
%         Coef. tangent needed
%         Coef. performance achieved
%Notes: The values of cu and ct are computed which generate
%        a match to solM's Cp
%        CpSol2 is typically passed for an entire rotation
%        along with Vrel, AoA

function [CuN CtP CtN Cp_p] = FindCuCtNeededToMatch(CpSol2,...
    TSR,C_R,Vrel,AoA,eaoa,ect,ecu)

%Define an anonymous function to define Cp
Cp=@(Ct,C_R,TSR,Vrel) 0.5.*C_R.*TSR.*Ct.*real(Vrel.*conj(Vrel));

%for each rotational position
%find the Ct needed
for ii=1:length(AoA)

```

```

        Ct_needed(ii)=fzero(@(Ct) Cp(Ct,C_R,TSR,Vrel(ii))-CpSol2(ii),-1);
        Cp_found(ii)=Cp(Ct_needed(ii),C_R,TSR,Vrel(ii));
    end

    %Compute the Ct lines from similar AoA
    for ii=1:length(ecu)
        for jj=1:length(AoA)
            ectf(ii,jj) = interp1(eaoa,ect(:,ii),(AoA(jj)));
        end
    end
    %for each rotational position
    %find the Cu needed based upon the Ct
    for ii=1:length(AoA)
        %set the extrapolation values to + or -
        if(Ct_needed(ii) <= 0)
            extpval=min(ecu);
        else
            extpval=max(ecu);
        end
        %Default case for nearest fit
        imethod='nearest';

        %Special case for linear fit
        if(AoA(ii) <= deg2rad(10) && AoA(ii) >= deg2rad(-10))
            extpval=0.01;
            imethod='linear';
        end

        %correct for all ectf being the same
        if(mean(ectf(:,ii)) >= ectf(1,ii)-eps && ...
            mean(ectf(:,ii)) <= ectf(1,ii)+eps)
            Cu_needed(ii)=0;
        else
            Cu_needed(ii)=interp1(ectf(:,ii),...
                ecu,Ct_needed(ii),imethod,extpval);
        end
        Ct_at_Cu(ii)=interpne(ect,[(AoA(ii)) Cu_needed(ii)],{eaoa,ecu});
        Cp_possible(ii)=Cp(Ct_at_Cu(ii),C_R,TSR,Vrel(ii));
    end
    CuN=Cu_needed;
    CtN=Ct_needed;
    CtP=Ct_at_Cu;
    Cp_p=Cp_possible;

```

A.8 FindPowerForCCEllipse

```

%Copyright (c) Jay P. Wilhelm 2010, All Right Reserved

%FindPowerForCCEllipse
%Purpose: Compute the power developed by a VAWT

```

```

%Inputs: Wind Velocity vector (m/s)
%         Rayleigh dist parameter, avg. wind speed (m/s)
%         Initial VAWT solidity
%         omega * R (rad/s * m)
%         Coeff. per of the VAWT vector (vs inv_tsr)
%         1/TSR
%         solidity list (to find max cp)
%Outputs: Power of the turbine
%         Power of the CC turbine
%         Percent increase using CC
%         Power curve of the turbine
%         Power curve of the CC turbine
%Notes: The EllipseCp parameter is a 3d vector of all
%        solidity ranges (0.01 to 0.2)

function [Pt,Pcc,per_inc,Pt_pure,Pcc_pure]=...
    FindPowerForCCEllipse(V,mu,sol0,wr,EllipseCp,inv_tsr,sol_list)
mu_r=mu;
wr_runat=wr;
sol_0=sol0;
for ii=1:length(inv_tsr)
    mxCp(ii)=max(EllipseCp(:,ii));
    mxCpLoc(ii)=find(EllipseCp(:,ii)==mxCp(ii));
end
mxSol=sol_list(mxCpLoc);
cp_s=mxCp;
x=find(cp_s<0);
cp_s(x)=0;
ft_ = fittype('pchipinterp');

Cp_CC=fit(inv_tsr(:),cp_s(:),ft_);
wr_CC=wr_runat;

KpList=EllipseCp;
soll=sol_list;
ft_ = fittype('pchipinterp');
rho_kgm3=1.2;

% mu_r=7;%m/s *3.2;
rD=rayleigh(V,mu_r);


```

%pre alloc the curve fits
[SolMx]=find(sol_list>=sol_0-eps,1,'First');
kpt = KpList(SolMx,:);
kp_fit=kpt(:);
kp_fit(find(kp_fit<0))=0;
Kp_BestSol = fit(inv_tsr(:),kp_fit,ft_);
wr_BestSol = wr_runat;

%Power of the wind
Pw=0.5.*rho_kgm3.*V.^3.*rD.*0.59;
Pw_t=trapz(V,Pw);

%Power of the turbine
Pt_pure=0.5.*rho_kgm3.*V.^3.*Kp_BestSol(V./wr_BestSol);
Pt=0.5.*rho_kgm3.*V.^3.*rD.*Kp_BestSol(V./wr_BestSol);

```


```

```

Pt=trapz(V,Pt);

%Power of the CC turbine
Pcc_pure=0.5.*rho_kgm3.*V.^3.*Cp_CC(V./wr_CC);
Pcc=0.5.*rho_kgm3.*V.^3.*rD.*Cp_CC(V./wr_CC);
Pcc=trapz(V,Pcc);
%percent increase using CC
per_inc=(Pcc-Pt)/Pt;

```

A.9 FindSolAndWRForCCEllipse

```

%Copyright (c) Jay P. Wilhelm 2010, All Right Reserved

%FindSolAndWRForEllipse
%Purpose: Compute the Cu needed to match solidity
%Inputs: mu, average wind speed (m/s)
%        Coeff. per of the VAWT vector (vs inv_tsr)
%        1/TSR
%        solidity list (to find max cp)
%Outputs: best solidity
%         best omega*R
%Notes: Parametric search of pre-computed Cp data

function [sol,wr]=FindSolAndWRForEllipse(mu,EllipseCp,inv_tsr,sol_list)
mu_r=mu;

for ii=1:length(inv_tsr)
    mxCp(ii)=max(EllipseCp(:,ii));
    mxCpLoc(ii)=find(EllipseCp(:,ii)==mxCp(ii));
end
mxSol=sol_list(mxCpLoc);
cp_s=mxCp;
x=find(cp_s<0);
cp_s(x)=0;
ft_ = fittype('pchipinterp');

Cp_CC=fit(inv_tsr(:),cp_s(:),ft_);

KpList=EllipseCp;
soll=sol_list;
or_list=[20:1:120];

ft_ = fittype('pchipinterp');
rho_kgm3=1.2;
V=(0:0.1:30)';

rD=rayleigh(V,mu_r);


```

%pre alloc the curve fits
for jj=1:length(soll)

```


```

```

    kpt = KpList(jj,:);
    kp_fit=kpt(:);
    kp_fit(find(kp_fit<0))=0;
    Kp_list{jj} = fit(inv_tsr(:),kp_fit,ft_);
end

for kk=1:length(or_list)
    for jj=1:length(soll)
        omega_r=or_list(kk);
        Kp=Kp_list{jj};
        Pw=0.5.*rho_kgm3.*V.^3.*rD.*0.59;
        Pt=0.5.*rho_kgm3.*V.^3.*rD.*Kp(V./omega_r);
        Pw_list(kk,jj)=trapz(V,Pw);
        Pt_list(kk,jj)=trapz(V,Pt);
    end
    p=0.5.*rho_kgm3.*V.^3.*rD.*Cp_CC(V./or_list(kk));
    Pcc_list(kk)=trapz(V,p);
end

%find the best case omegaR & solidity
[orMx,SolMx]=find(Pt_list>=0.999.*max(max(Pt_list)),1,'First');
wr=or_list(orMx);
sol=soll(SolMx);

```